© 2023 Yiren Wang

EXPLOITING MONOLINGUAL DATA FOR NEURAL MACHINE TRANSLATION

BY

YIREN WANG

DISSERTATION

Submitted in partial fulfillment of the requirements
for the degree of Doctor of Philosophy in Computer Science
in the Graduate College of the
University of Illinois Urbana-Champaign, 2023

Urbana, Illinois

Doctoral Committee:

Professor ChengXiang Zhai, Chair
Professor Julia Hockenmaier
Professor Heng Ji
Doctor Hany Hassan Awadalla, Microsoft

ABSTRACT

Neural Machine Translation (NMT) has made rapid progress in recent years and achieved remarkable translation quality. NMT systems heavily rely on large-scale and high-quality parallel training data of source and target languages (bitext), which is limited and costly to collect. In the meantime, large amount of unlabeled monolingual data is available in different languages. While the target-side monolingual data has been proven useful through Back Translation, the source-side data is less investigated. Effectively exploiting both source-side and target-side monolingual data could not only help boost NMT performance, but also improve data efficiency. In this thesis, we focus on exploring novel approaches that effectively utilize monolingual data to improve the translation quality of NMT models.

First, we exploit the potential of monolingual data with dual learning. Dual learning effectively utilize both bitext and monolingual data by leveraging the primal-dual structure of artificial intelligence tasks to generate informative feedback signals to regularize training. We extend the standard dual learning framework by introducing multiple primal and dual models, and propose the Multi-Agent Dual Learning (MADL) to further boost data utilization and translation quality. We show the effectiveness of our proposed approach on multiple benchmark NMT tasks, including both supervised and unsupervised NMT.

Second, we study the effect of forward and back translation data at scale, and propose a large-scale noisy training pipeline to effectively leverage both of them. First, we generate synthetic bitext with both forward and back translation. Next, we train the NMT model on a noisy version of this synthetic corpus, where each source sentence is randomly corrupted. Finally, the model is fine-tuned on the genuine bitext and a clean, high-quality subset of the synthetic bitext without any noise. With our proposed strategy, we achieve the state-of-the-art performances on various benchmark translation tasks.

Third, we explore more general strategies to utilize monolingual data for not only high-resource language pairs in bilingual NMT, but also low-resource language pairs and multilingual NMT as well. We propose a multi-task learning (MTL) framework, which jointly trains the model with the translation task on bitext data, and two auxiliary self-supervised learning tasks on the source-side and target-side monolingual data respectively. We show that our approach can effectively improve translation quality in the multilingual setting for both high-resource and low-resource languages. We also demonstrate the effectiveness of our method over pre-training approaches for both NMT and cross-lingual transfer learning natural language understanding tasks.

To my parents, for their love and support.

ACKNOWLEDGMENTS

First and foremost, I am deeply grateful to my advisor Professor ChengXiang Zhai for his invaluable supervision and tutelage during the course of my Ph.D. I sincerely appreciate his passion and expertise in research, his insightful feedback and comments in the creation of this thesis, and his unlimited support throughout the whole process. In addition to that, I am particularly thankful for his vision to think "big" and his encouragement in finding inspiration for my research work as well as the possibilities in my life.

Furthermore, I would like to express my sincere gratitude to my committee members: Professor Julia Hockenmaier, Professor Heng Ji and Dr. Hany Hassan Awadalla. I had the privilege to attend Julia's class in natural language processing in the early stage of my Ph.D. I am very grateful for her openness in offering me insightful advice and feedback. I had the pleasure of discussing with Heng and Ph.D. students in her group soon after she joined University of Illinois at Urbana-Champaign (UIUC). I have greatly benefited from the inspiring discussion and invaluable suggestions. I had the privilege to work with Hany since my internship at Microsoft. I deeply appreciate his guidance and feedback in my research projects, as well as his support and advice in the early stage of my professional career.

I have greatly benefited from many other members of academia and industry, and I would like to use this opportunity to express my sincere thanks (in alphabetic order): Professor Xueqi Cheng (Chinese Academy of Sciences), Dr. Orhan Firat (Google), Professor Di He (Peking University), Dr. Hongzhao Huang (Meta), Dr. Tie-yan Liu (Microsoft), Professor Qiaozhu Mei (University of Michigan), Dr. Arul Menezes (Microsoft), Dr. Fuchun Peng (Meta), Dr. Tao Qin (Microsoft), Professor Maosong Sun (Tsinghua University), Dr. Lijun Wu (Microsoft), Dr. Yingce Xia (Microsoft) and Wei Zhang (Visa Research).

I would also like to thank all my collaborators and friends that I had the pleasure of working with during my Ph.D. journey (in alphabetic order): Dr. Huan Gui, Dr. Shan Jiang, Dr. Sean Massung, Dr. Shubhra Kanti Karmaker Santu, Dr. Dominic Seyler, Dr. Xiaolong Wang, Jinfeng Xiao, and Dr. Yinan Zhang.

Finally, I would like to extend my sincere gratitude towards my husband Fei Tian, my parents and my family for their love and support.

TABLE OF CONTENTS

CHAPTER 1 INTRODUCTION 1
1.1 Improving NMT with Monolingual Data 1
1.2 Thesis Outline 4

CHAPTER 2 BACKGROUND 5
2.1 Neural Machine Translation 5
2.2 Multilingual Neural Machine Translation 7
2.3 Improving NMT with Monolingual Data 8

CHAPTER 3 UTILIZING MONOLINGUAL DATA WITH DUAL LEARNING . . 12
3.1 Introduction 12
3.2 Framework 14
3.3 Adaptation to Neural Machine Translation 16
3.4 Experiments 18
3.5 Application to WMT2019 News Translation Task 27
3.6 Extention to Image Translation 29
3.7 Summary 34

CHAPTER 4 EXPLOITING MONOLINGUAL DATA AT SCALE 35
4.1 Introduction 35
4.2 Background 36
4.3 Approach 39
4.4 Experimental Setup 41
4.5 Results 43
4.6 Analysis 45
4.7 Combination with MADL 50
4.8 Summary 53

CHAPTER 5 UTILIZING MONOLINGUAL DATA WITH MULTI-TASK LEARNING 54
5.1 Introduction 54
5.2 Background 55
5.3 Approach 57
5.4 Experimental Setup 60
5.5 Results 62
5.6 Analysis 66
5.7 Summary 72

CHAPTER 6 CONCLUSIONS AND OUTLOOK 73
REFERENCES 76

CHAPTER 1: INTRODUCTION

Machine Translation, which automatically translates text from one language to another, has made rapid progress in recent years. Neural Machine Translation (NMT) is an approach that models the entire machine translation process in an end-to-end manner via one big artificial neural network. Achieving remarkable translation quality and the state-of-the-art results, NMT has replaced the prior linguistically informed Statistical Machine Translation (SMT) approaches, and has become the new standard of machine translation in both research and industry.

NMT benefits from the three pillars of the deep learning era, that is, the big data, big model, and big computing. Firstly, huge amount of data in natural language from different sources are available to use, including not only the traditional articles and books, but also long documents from the webpages, dialogues from the social networks, as well as data collected with new mechanisms such as crowd-sourcing. Big data makes it possible to train complex neural network models on large-scale data sources, and helps to improve the model quality and promote generalization. Secondly, the introduction of deep and wide neural networks with tens and hundreds of layers, and billions of parameters makes it possible to well leverage the big data and make breakthroughs. In the meantime, the big computing resources, from the CPU clusters to GPU and TPU clusters, makes it possible to train the big models with the big data.

While deep learning has brought big progress and opportunities to the field of NMT, it is also facing lots of challenges. One of them is the big data challenge. Deep learning highly relies on huge amount of human-labeled training data. Specifically in the context of NMT, it refers to the bitext training data, i.e., the parallel sentence or document pairs in the source and target languages respectively. In practice, the annotated bitext data are limited, and human labeling is very expensive to collect. Beyond the cost, it can be very hard, if possible at all, to obtain large-scale labeled data in some less popular languages and domains.

1.1 IMPROVING NMT WITH MONOLINGUAL DATA

While labeled bitext data is limited and hard to obtain, the unlabeled monolingual data, i.e., sentences or documents in a single language, is largely available and cheap to collect for most languages. Effectively utilizing monolingual data can greatly ease the data challenge and further push the development of NMT.

In this thesis, we focus on exploiting monolingual data to improve the performance of

NMT systems. Specifically, we explore the following two research questions:

- How to effectively utilize the large-scale monolingual data to help boost the performance of NMT models?
- Given the bitext and monolingual data we have, how to improve data efficiency?

To answer these questions, in this thesis, we introduce several training strategies and frameworks as solutions to utilize monolingual data in both source and target languages.

NMT is an artificial intelligence task with structural duality, namely, any NMT task has a dual task, e.g., English-to-German translation and German-to-English translation. We formulate them as a primal task that learns the mapping from domain $\mathcal{X}$ to domain $\mathcal{Y}$ (e.g. English-to-German translation), and a dual task that learns the reversed mapping (e.g. German-to-English translation). The primal and dual tasks form a closed loop and can generate informative feedback signals for each other. Dual learning [1] is proposed to leverage such feedback signals to regularize training. In Chapter 3, we extend the dual learning framework by introducing multiple primal and dual models into the learning system, and propose the Multi-Agent Dual Learning (MADL) framework to further improve data utilization and translation quality. The additionally introduced models in the same direction have similar capability and certain level of diversity to map one domain to the other domain. They work together to provide more reliable and robust feedback signals, and play the role of facilitating the training. We show that MADL can effectively improve translation quality and data efficiency. In particular, MADL set a new record on IWSLT2014 German-to-English translation, achieve the state-of-the-art performance on the recent WMT2018 English-to-German translation. MADL is also one of the key techniques used in Microsoft Research Asia's systems [2] for WMT2019 shared task in news translation and ranked first in Enlish-to-German, German-to-English, French-to-German and German-to-French tasks.

In the context of NMT, unlabeled monolingual data consists of two categories based on their language domain, i.e., the source-side and the target-side. The target-side monolingual data has been proven effective with the development of Back Translation [3] techniques. Back translation leverages a target-to-source translation model to translate the target-side monolingual data into source language. In this way, the target-side monolingual data can be effectively utilized as pseudo bitext with pairs of synthetic source sentences and genuine target sentences to augment the training data. In the meantime, the source-side monolingual data are less investigated. Symmetrically, the forward translation on the source-side monolingual data can be performed to general pseudo bitext. However, the conventional wisdom is that using synthetic forward translated data would harm the target language model. Therefore, the source-side monolingual data has been largely ignored.

To fully utilize both source and target-side monolingual data effectively, in Chapter 4, we investigate the effects of both forward and back translation data through a systematic empirical study. We introduce a large-scale noisy training pipeline to fully utilize monolingual data in both source and target languages at scale to boost the accuracy of NMT. First, we generate synthetic bitext with both forward and back translation. An NMT model is then trained on a noisy version of the synthetic bitext corpus, where each source sentence is randomly corrupted. Finally, the model is fine-tuned on the genuine bitext and a clean, high-quality subset of the synthetic bitext without any noise. We show that our training strategy achieves state-of-the-art results on English↔German translations in WMT2016, 2017 and 2018, as well as German→French translations in WMT2019.

While our proposed large-scale noisy training strategy can effectively utilize monolingual data in both languages to boost the capability of NMT models, several limitations remain for leveraging monolingual data through forward and back translations. As well-trained NMT models are required to generate forward or back translations for each language pair, it is computationally very expensive to scale. Moreover, it is less applicable to low-resource language pairs without adequate bitext data.

To overcome such limitations, in Chapter 5, we introduce a multi-task learning framework to effectively utilize monolingual data with self-supervised learning techniques. In the proposed framework, the NMT model is jointly trained with the translation task on bitext data, and two self-supervised learning tasks, masked language modeling and denoising auto-encoding on the source-side and target-side monolingual data respectively. We investigate the multi-task framework in the multilingual NMT setting, which is a new paradigm for NMT and has drawn great research and commercial attention in recent years. We present two simple yet effective scheduling strategies for the multilingual and multi-task framework. Our results on multilingual NMT systems with 10 language pairs demonstrate that the proposed approach can effectively improve the translation quality for both high-resource and low-resource languages with large margin, achieving significantly better results than the individual bilingual models. It also shows efficacy in the zero-shot setup for language pairs without bitext training data.

This thesis was completed before ChatGPT was released. The emergence of the Large Language Models (LLMs) has no doubt opened more opportunities for using such LLMs to perform machine translation. The contributions of this thesis are mostly orthogonal in the sense that they can be used to enhance the performance of LLMs for NMT.

1.2 THESIS OUTLINE

The remainder of this thesis is organized as follows. In Chapter 2, we review the background and related literature of NMT. From Chapter 3 to Chapter 5, we introduce our three lines of approaches to effectively utilize monolingual data to improve the performance of NMT models respectively. Specifically, Chapter 3 introduces a novel Multi-Agent Dual Learning framework to improve data efficiency with dual learning. Chapter 4 presents a systematic study on forward and back translation at scale, and introduces our large-scale noisy training strategy. Chapter 5 introduces a multi-task learning framework with self-supervised learning, which can effectively utilize monolingual data for multilingual neural machine translation. Finally, We conclude this thesis by summarizing our contributions and and discussing open questions in Chapter 6.

CHAPTER 2: BACKGROUND

2.1 NEURAL MACHINE TRANSLATION

Neural Machine Translation (NMT) is a sequence generation problem which can be formulated as follows: There are two domains $\mathcal{X}$ and $\mathcal{Y}$, associated with vocabulary $\mathcal{V}_x$ and $\mathcal{V}_y$ respectively. The vocabulary here consists of tokens which are the basic building blocks of the data. Any data sample $x \in \mathcal{X}$ is an ordered combination of tokens in $\mathcal{V}_x$, that is, $x = (x_1, x_2, \cdots, x_{l_x})$, where $x_i \in \mathcal{V}_x$ for any $i \in [l_x]$, and l_x is the number of word tokens in x. Similarly, we have $y = (y_1, y_2, \cdots, y_{l_y})$, $y_j \in \mathcal{V}_y$ where $j \in [l_y]$. Our target is to learn a mapping $f : \mathcal{X} \mapsto \mathcal{Y}$ that can map any $x \in \mathcal{X}$ to the corresponding $y \in \mathcal{Y}$. Specifically, in NMT tasks, $\mathcal{X}$ and $\mathcal{Y}$ are collections of sequences of words in two different languages.

Most NMT systems adopt the structure of the encoder-decoder framework [4], where the input sequence $x = (x_1, x_2, \cdots, x_{l_x})$ is firstly processed by the encoder to get a sequence of hidden representations $z = (z_1, z_2, \cdots, z_{l_x})$. z is then fed into the decoder to generate an output sequence $y = (y_1, y_2, \cdots, y_{l_y})$. The encoder and decoder can be specialized using different neural architectures including Long Short-Term Memory Networks (LSTM) [5], Convolutional Neural Network (CNN) [6], and Transformer [7], among which Transformer is the most recent and state-of-the-art architecture for NMT.

A Transformer model consists of three types of basic layers:

- *Self-attention layer*: Self attention $F_S(Q, K, V)$, where the query Q, key K and value V are all the hidden representations from the previous layer, generates an adaptive weighted sum of the input hidden representations.

- *Feed-forward layer*: Feedforward $F_N(h)$ applies non-linear transformations to the hidden vector h.

- *Source-to-target attention layer*: Source-to-target attention $F_C(Q, K, V)$ takes in the hidden representations of the previous layer as Q and context vectors from the encoder as K and V, and adaptively aggregates the information from the encoder side. F_C exists in decoder only.

Each basic layer is associated with layer normalization [8], residual connection [9] and dropout [10]. In a Transformer, the encoder and decoder are stacks of $F_N \circ F_S$ and $F_N \circ F_C \circ F_S$, where $\circ$ represents the function cascading. The core component of the Transformer is the attention module $\texttt{attn}(q, V)$, where $q \in \mathbb{R}^d$ and V consists of $|V|$ elements $\{v_1, v_2, \cdots, v_{|V|}\}$

with $v_i \in \mathbb{R}^d$. Mathematically,

$$\texttt{attn}(q, V) = \sum_{i=1}^{|V|} \alpha_i v_i;\ \alpha_i \propto \exp(W_q q + W_v v_i) \tag{2.1}$$

where $\alpha_i \geq 0$, $\sum_{i=1}^{|V|} \alpha_i = 1$, and W's are the parameters to be learned.

An input $x \in \mathcal{X}$ is first mapped into a representation $H^0 = (x_{x_1} E_x, x_{x_2} E_x, \cdots, x_{x_{l_x}} E_x)$ through embedding matrix. We use H^l to denote $(h_1^l, h_2^l, \cdots, h_{l_x}^l)$ where the superscript $l \in \{0, 1, \cdots, L\}$ represents the layer id with the 0'th layer as the embedding layer, and $h_i \in \mathbb{R}^d$ for any $i \in [l_x]$.

The encoder works in a recursive way that $H^l = \texttt{enc}_l(H^{l-1})$, $l \in [L]$. $\texttt{enc}_l$ consists of two sub-layers: (1) a self-attention layer that provides an adaptive weighted sum of the hidden representations at each time step from the $l-1$'th layer; and (2) a non-linear layer F_N where $F_N(x) = \max(xW_1 + b_1, 0)W_2 + b_2$, where max is an element-wise operator, W's and b's are the parameters to be learned. Then we have,

$$\texttt{enc}_{l+1} = F_N(\texttt{attn}(h_i^l, H^l)) \tag{2.2}$$

The encoder will eventually output H^L.

The decoder works in a similar way as the encoder with the only difference of leveraging an additional part to deal with the information from the encoder. Each layer in the decoder has three sub-layers: a self-attention layer F_S, source-to-target attention layer F_C and a non-linear layer F_N. F_S and F_C with different parameters. Let s_t^l denote the output of the l'th layer in the decoder at times-tep $t(> 0)$ and let S_t^l denote $\{s_1^l, \cdots, s_t^l\}$, $l \in [L]$. Specially, $S_t^0 = (y_{y_1} E_y, y_{y_2} E_y, \cdots, y_{y_t} E_y)$. Mathematically we have,

$$\texttt{dec}_l(s_t^{l-1}, S_t^{l-1}, H^L) = F_N(F_C(F_S(s_t^{l-1}, S_t^{l-1}), H^L)). \tag{2.3}$$

Finally, s_t^L will be fed into a softmax layer to get the probabilities for each word in $\mathcal{V}_y$.

NMT models are usually trained in an end-to-end manner with parallel training data of source and target languages (bitext for short). Given a bitext corpora D_B of sentence pairs (x, y), i.e., $D_B = \{(x, y)\}$, NMT is trained with the following cross-entropy loss (i.e. negative log-likelihood) $\mathcal{L}_{MT}$:

$$\mathcal{L}_{MT} = \mathbb{E}_{(x,y)\sim D_B}[-logP(y|x)] \tag{2.4}$$

[11] proposed the state-of-the-art unsupervised NMT algorithm. Similar to the standard NMT model with bilingual training data, an unsupervised NMT model is also based on

the encoder-decoder architecture. The source sentences and target sentences are mapped into a same vocabulary using BPE techniques. The shared embedding is pretrained with fastText [12] to get good initial values. The proposed model can handle both source-to-target translation and target-to-source translation like [13]. The training loss usually consists of two parts, a language model loss and a back-translation loss. Let $P_{\mathcal{X}\to\mathcal{Y}}$ denote the translation from space $\mathcal{X}$ to space $\mathcal{Y}$, and so for the other similar notations.

- Language model loss, implemented by a denoising autoencoder. Mathematically,

$$\mathcal{L}_{LM} = \mathbb{E}_{x\sim\mathcal{X}}[-\log P_{\mathcal{X}\to\mathcal{X}}(x|C(x))] + \mathbb{E}_{y\sim\mathcal{Y}}[-\log P_{\mathcal{Y}\to\mathcal{Y}}(y|C(y))] \tag{2.5}$$

  where $C(\cdot)$ is a noising model that randomly drops or swaps words.

- Back translation loss, implemented by back-translating the monolingual data and feeding into the reversed models. Mathematically,

$$\mathcal{L}_{BT} = \mathbb{E}_{x\sim\mathcal{X}}[-\log P_{\mathcal{Y}\to\mathcal{X}}(x|\hat{y}(x))] + \mathbb{E}_{y\sim\mathcal{Y}}[-\log P_{\mathcal{X}\to\mathcal{Y}}(y|\hat{x}(y))] \tag{2.6}$$

  in which $\hat{y}(x) = \arg\max_{u\in\mathcal{Y}} P_{\mathcal{X}\to\mathcal{Y}}(u|x)$ and $\hat{x}(y) = \arg\max_{v\in\mathcal{X}} P_{\mathcal{Y}\to\mathcal{X}}(v|y)$. Note that the four $P_{\ldots}$'s are implemented in a single encoder-decoder based model, where each translation task has a different "task embedding", i.e., a learnable vector indicating the translating directions. We will discuss more details into back translation in Section 2.3.1

2.2 MULTILINGUAL NEURAL MACHINE TRANSLATION

Multilingual Neural Machine Translation (MNMT), which leverages a single NMT model to handle the translation of multiple languages, has drawn research attention in recent years [13, 14, 15, 16, 17]. MNMT is appealing since it greatly reduces the cost of training and serving separate models for different language pairs [13]. It has shown great potential in knowledge transfer among languages, improving the translation quality for low-resource and zero-shot language pairs [17, 18, 19].

Multilingual Neural Machine Translation (MNMT) aims to train a single translation model that translates between multiple language pairs at the same time [13, 15], without changing the traditional NMT model architecture.

Previous works in MNMT explored the model architecture design with different parameter sharing strategies, such as partial sharing with shared encoder [14, 20], shared attention [15],

task-specific attention [21], and full model sharing with language identifier [13, 16, 17]. There are also extensive studies on representation sharing that shares lexical, syntactic or sentence level representations across different languages [18, 22, 23].

Dong et al. [14] proposes to translate one source language into several different target languages with a unified translation model under the multi-task learning framework. Specifically, the encoder of the NMT model is shared across different language pairs, whereas each target language has a separate decoder. The key motivation is that many languages are closely related on the semantic or syntactic levels, even when they are differ lexically. The shared encoder enables the model to learn the semantic and structured predictive representations of the source language across all the translation tasks into different languages, which are otherwise hard to learn with only a small amount of data.

Arivazhagan et al. [17] introduces an massively MNMT model handling 103 languages trained on over 25 billion examples. Their system shows remarkable capability in transfer learning among different language pairs, and thus is able to significantly boost the translation quality of low-resource languages, while maintaining high quality of high-resource languages.

However, existing MNMT systems mainly rely on bitext for training, and the largely available monolingual data has not been effectively leveraged.

2.3 IMPROVING NMT WITH MONOLINGUAL DATA

NMT heavily relies on large-scale parallel dataset for training. To augment the limited bilingual data, there are plenty of works attempt to leverage the monolingual data to help the training, which includes back translation [3], dual learning [24] and self learning [25].

2.3.1 Back Translation

Back translation (BT) [3], which translates the target-side monolingual data into source language to generate pseudo bitext for training, has been one of the most effective approaches in NMT. Specifically, an additional target-to-source NMT model is firstly trained on the genuine bitext. This model will be used to translate target-side monolingual sentences $y_m \sim \mathcal{Y}$ into the source domain $x' \sim \mathcal{X}$. The translation output and the target-side monolingual data is paired as synthetic parallel corpus $D_{BT} = \{(x', y_m)\}$ to augment the original bitext dataset D_B in order to further train the source-to-target NMT model.

Since BT is widely acknowledged and effective to improve the NMT model, there have been several works investigating BT from different views [26, 27, 28], and different variations of BT have been proposed [24, 29, 30].

Poncelas et al. [26] investigates the unknown factors regarding the actual effects of BT data on the translation capability of an NMT model. They study on how using the back-translated data as a training corpus, either used separately or combined with human-annotated bitext, affects the performance of an NMT model. They use incrementally larger amounts of BT data for training, and find that while larger amounts of synthetic BT data brings improvements, the performance of NMT model starts to drop when the balance is tipped too far in favor of the synthetic data.

Burlot and Yvon [27] conducts a systematic study of BT. They analyze the accuracy impact from the quality of the BT data, the alternative uses of BT data to give explanations of why BT works. They show that the quality of BT matters for NMT. They also introduce new data simulation techniques that can generates synthetic data as good as low quality BT, yet is cheaper to compute. Specifically, they propose to use noised target sentence, or leverage GANs to generate the pseudo-source sentences, when there is no high quality BT data available.

Cotterell and Kreutzer [28] provides an interpretation of back translation as approximate inference in a generative model of bitext and give a new algorithm. However, above studies are all based on the small-scale monolingual data, therefore it remains unclear in the large-scale setting.

Edunov et al. [29] firstly provides an extensive analysis of the back translation at scale. They investigate the different synthetic data generation methods, and also compare the different combination of the synthetic data with bitext. They finally build a strong system with millions of target-side monolingual sentences.

2.3.2 Dual Learning

Dual learning [24] extends the BT approach to train NMT systems in both translation directions. When jointly training the source-to-target and target-to-source NMT models, the two models can provide back translated data for each other direction and perform multi-rounds BT. This strategy is also successfully adopted to build the unsupervised translation system [31].

The basic idea of dual learning is to leverage the duality between the two tasks as the feedback signal to regularize training. The regularization has been implemented with different ways in existing works, such as maximizing the likelihood of data reconstruction [24], constraining joint probabilistic consistency [1], and encouraging the model-level information sharing [32].

Among different variations of dual learning, the principle of likelihood maximization of

data reconstruction has been mostly adopted [24, 33], in which dual learning is formulated as a two-agent game: One agent targets at learning the mapping $f : \mathcal{X} \mapsto \mathcal{Y}$, while the other learning the mapping of opposite direction $g : \mathcal{Y} \mapsto \mathcal{X}$. Here $\mathcal{X}$ and $\mathcal{Y}$ are two domains. In dual learning, $x \in \mathcal{X}$ is first mapped to $\hat{y} \in \mathcal{Y}$ through f and then reconstructed to $\hat{x} \in \mathcal{X}$ by $g(\hat{y})$. The distortion between x and $\hat{x}$, denoted as $\Delta_x(x, \hat{x})$, is used as the feedback signal to regularize training. Similar reconstruction error can also be constructed in $\mathcal{Y}$ and further used for training.

2.3.3 Source-side Monolingual Data

While target-side monolingual data has been demonstrated effective through back translation, only a few attempts have been made working on using the source-side monolingual data [25, 34].

Zhang and Zong [25] proposes self learning approach to generate the synthetic data for the source-side monolingual data, which is a semi-supervised method. Wu et al. [34] leverages the source-side monolingual data to train the NMT system by learning reward function in a reinforcement learning framework. Gulcehre et al. [35] integrates the hidden states from the target language model into the NMT decoder to improve the accuracy.

2.3.4 Self-supervised Learning

Self-supervised pre-training approaches, which train the model with denoising learning objectives on the large-scale monolingual data, have achieved remarkable performances in many NLP applications, such as question answering [36], natural language inference [37], paraphrasing [38], etc. Many different denoising objectives have been designed to train the neural networks on large-scale unlabeled text [39, 40, 41, 42, 43].

Radford et al. [39] uses the standard language modeling as the only self-supervised learning objective in the pre-training of the GPT model. Given an unlabeled corpus D_M, language modeling maximize the likelihood:

$$\mathcal{L}_{LM} = \sum_i logP(x_i|x_{i-k}, ..., x_{i-1}; \Theta) \tag{2.7}$$

where k is the size of the context window, and the conditional probability P is modeled using a neural network with parameter Θ.

Masked Language Model (MLM) is a cloze-style task firstly proposed in the pre-training of BERT [40]. In MLM, sentences with tokens randomly masked are fed into the model, and

the model attempts to predict the masked tokens based on their context. 15% tokens in each sequence are randomly masked during the pre-training of BERT. Among the masked tokens, 15% of them will be replaced with a special `[MASK]` token 80% of the time, a random token 10% of the time, or remain unchanged 10% of the time. Compared with traditional left-to-right or right-to-left language modeling, MLM is beneficial for learning deep bidirectional representations.

Denoising Auto-Encoding (DAE) [44] has been demonstrated to be an effective strategy for unsupervised NMT [11, 45]. Given a monolingual corpus D_M and a stochastic noising model C, DAE minimizes the following reconstruction loss:

$$\mathcal{L}_{DAE} = \mathbb{E}_{x \sim D_M}[-\log P(x|C(x))] \tag{2.8}$$

Different noising scheme can be applied for the noising model C. Token masking [42] randomly sample tokens and replace them with a special `[MASK]` token. Token deletion [11] randomly sample and delete tokens from the input. Token shuffle [11] slightly shuffle the input sentence. Text infilling [42, 46] samples a number of text spans, with with span lengths drawn from a Poisson distribution ($\lambda = 3$). Each text span is replaced with a single `[MASK]` token. And a 0-length span correspond to the insertion of a `[MASK]` token.

While self-supervised learning has demonstrated remarkable performances in various NLP applications, catastrophic forgetting effect [47], where finetuning on a task leads to degradation on the main task, limits the success of continuing training NMT on models pre-trained with monolingual data. Furthermore, the separated pre-training and finetuning stages make the framework less flexible to introducing additional monolingual data or new languages into the NMT system.

CHAPTER 3: UTILIZING MONOLINGUAL DATA WITH DUAL LEARNING

Dual learning has attracted much attention in machine learning, computer vision and natural language processing communities. The core idea of dual learning is to leverage the duality between the primal task (mapping from domain $\mathcal{X}$ to domain $\mathcal{Y}$) and dual task (mapping from domain $\mathcal{Y}$ to $\mathcal{X}$) to boost the performances of both tasks. Existing dual learning framework forms a system with two agents (one primal model and one dual model) to utilize such duality. In this chapter, we extend the dual learning framework by introducing multiple primal and dual models, and propose *Multi-Agent Dual Learning (MADL)* framework. Experiments on various neural machine translation tasks with different language pairs and training data scales demonstrate the effectiveness of the MADL framework. We further apply the framework to tasks beyond neural machine translation and show its effectiveness in image translation tasks, which demonstrate the generality of MADL.

3.1 INTRODUCTION

Motivated by the dual nature of many tasks, e.g., English-to-German vs. German-to-English in machine translation, photo-to-Monet vs. Monet-to-photo in image translation, and speech recognition vs. speech synthesis, dual learning is proposed [24] and applied to many applications including neural machine translation (NMT) [1, 24, 48], image-to-image translation [33, 49, 50], question answering [51, 52] and image captioning [53]. The basic idea of dual learning is to leverage the duality between the two tasks as the feedback signal to regularize training. The regularization has been implemented with different ways in existing works, such as maximizing the likelihood of data reconstruction [24], constraining joint probabilistic consistency [1], and encouraging the model-level information sharing [32]. Among them, the principle of likelihood maximization of data reconstruction has been mostly adopted [24, 33], in which dual learning is formulated as a two-agent game: One agent targets at learning the mapping $f : \mathcal{X} \mapsto \mathcal{Y}$, while the other learning the mapping of opposite direction $g : \mathcal{Y} \mapsto \mathcal{X}$. Here $\mathcal{X}$ and $\mathcal{Y}$ are two domains. In dual learning, $x \in \mathcal{X}$ is first mapped to $\hat{y} \in \mathcal{Y}$ through f and then reconstructed to $\hat{x} \in \mathcal{X}$ by $g(\hat{y})$. The distortion between x and $\hat{x}$, denoted as $\Delta_x(x, \hat{x})$, is used as the feedback signal to regularize training. Similar reconstruction error can also be constructed in $\mathcal{Y}$ and further used for training.

In such a two-agent game, g and f can be viewed as an evaluator for each other. g is used to evaluate the quality of $\hat{y}$ generated by f and return the feedback signal $\Delta_x(x, g(\hat{y}))$ back to f, and vice versa. The quality of such evaluation plays an important role to improve

the training of the mapping functions. In the current dual learning framework, only one agent g is used to evaluate and provide feedback signals to the mapping function f in the other direction. In this chapter, we introduce multiple agents into the learning system to further exploit the potential of dual learning. The agents in the same direction have similar capability and certain level of diversity to map one domain to the other domain, i.e., $\mathcal{X} \to \mathcal{Y}$ or $\mathcal{Y} \to \mathcal{X}$. Different agents can be obtained by training multiple f's and g's independently with different random seeds for initialization and data access order. Then for the output of each f (or g), multiple g's (or f's) will provide feedback signals. Intuitively, more agents can lead to more reliable and robust feedback, just like the majority voting of multiple experts, and it is expected to achieve better final performance. We name this new dual learning framework with multiple agents as *Multi-Agent Dual Learning (MADL)*[1].

Although multiple agents are involved in multi-agent dual learning, we still focus on training two mappings $f_0 : \mathcal{X} \to \mathcal{Y}$ and $g_0 : \mathcal{Y} \to \mathcal{X}$, similar to the traditional two-agent dual learning. The additionally introduced agents play the role of facilitating the training of f_0 and g_0. The parameters of the additional agents are fixed during training and have no part to play during testing/inference. Therefore, the inference of multi-agent dual learning is of the same complexity as it is of the standard two-agent dual learning. More precisely, apart from the two agents f_0 and g_0, we further employ $2(N-1)$ additional agents, $N \geq 2$, which are pre-trained before optimizing f_0 and g_0 and subsequently held constant. Among them, $N-1$ agents are leveraged to model the mapping $f_i : \mathcal{X} \mapsto \mathcal{Y}$ and the others $N-1$ agents model $g_i : \mathcal{Y} \mapsto \mathcal{X}$, $i \in \{1, 2, \cdots, N-1\}$[2]. All these $2(N-1)$ agents are used to facilitate the training of f_0 and g_0 in a similar manner as the standard dual learning with minor revisions: To help the training of f_0, for any $x \in \mathcal{X}$, we first map x to $\hat{y}$ by f_0, build the feedback signals by adaptively summing $\Delta_x(x, g_i(\hat{y}))$ for any $i \in \{0, 1, \cdots, N-1\}$, and then regularize training with the feedback signals. The training of g_0 is performed similarly.

We conduct extensive experiments on various English↔German translation tasks with different scales of bilingual data. On IWSLT 2014 translation task with $153K$ bilingual data and no monolingual data, we achieve 29.52 and 35.44 BLEU scores for English→German and German→English translations respectively, setting a new record on this task. On WMT 2014 translation task with $4.5M$ bilingual sentence, we achieve 30.05/31.03 and 33.32/35.64 BLEU scores without/with monolingual data, which are the best results of the same set-

[1]Although with the same term, the "multi-agent" or "agent" here has no relationship with multi-agent reinforcement learning

[2]The numbers of agents served for two mapping directions are not necessarily the same. For the tasks where the two domains $\mathcal{X}$ and $\mathcal{Y}$ are fully symmetric, e.g., the language spaces in NMT and image collections in image-to-image translation, learning the two mappings are of similar difficulty. Therefore, we use the same number of agents to assist training and leave the study of asymmetric setting to the future work.

tings. On the WMT 2018 English→German translation task, we set a new record of 49.61 BLEU score, improving the previous best system by over 1.3 BLEU score. On WMT 2016 unsupervised NMT task, where no bilingual data is available and only $50M$ monolingual data is provided, we achieve 19.26/23.85 BLEU scores, outperforming all the previous pure NMT based systems.

We further explore the generality of the proposed framework and apply MADL to image translation. Specifically, we work on two groups of tasks for image-to-image translation, paint↔photo translations (paint including Van Gogh, Monet, Ukiyo-e and Cezanne), and cityscapes label↔photo translation. Experimental results show that we can generate better images than CycleGAN.

3.2 FRAMEWORK

In this section, we mathematically formulate the multi-agent dual learning framework, and compare it with previous learning settings.

Consider two domains of interests, $\mathcal{X}$ and $\mathcal{Y}$. Let $\mathcal{D}_x$ and $\mathcal{D}_y$ denote the collection of training data from $\mathcal{X}$ and $\mathcal{Y}$ respectively, $\mathcal{D}_x \subset \mathcal{X}$, $\mathcal{D}_y \subset \mathcal{Y}$. We aim to learn two agents $f_0 : \mathcal{X} \mapsto \mathcal{Y}$, and $g_0 : \mathcal{Y} \to \mathcal{X}$. Without loss of generality, we refer to learning f_0 and g_0 as the primal task and the dual task respectively. Let $\Delta_x(x, x')$ be a mapping $\mathcal{X} \times \mathcal{X} \to \mathbb{R}$, representing the dissimilarity/distance/error between x and x' where $x, x' \in \mathcal{X}$, and $\Delta_y(y, y')$ denote the dissimilarity between the two elements y and y' in space $\mathcal{Y}$. The standard dual learning loss [24] is formulated as

$$\mathcal{L}_{Dual} = \frac{1}{|\mathcal{D}_x|} \sum_{x \in \mathcal{D}_x} \Delta_x(x, g_0(f_0(x))) + \frac{1}{|\mathcal{D}_y|} \sum_{y \in \mathcal{D}_y} \Delta_y(y, f_0(g_0(y))), \tag{3.1}$$

where $|\mathcal{D}_x|$ and $|\mathcal{D}_y|$ denote the number of elements in $\mathcal{D}_x$ and $\mathcal{D}_y$. In a multi-agent dual learning framework, as a prerequisite there are $N-1$ pretrained primal models $f_i : \mathcal{X} \to \mathcal{Y}$ and $N-1$ dual models $g_i : \mathcal{Y} \to \mathcal{X}$, $i = \{1, 2, \cdots, N-1\}$.

Each f_i is pretrained by minimizing the cross entropy loss $\mathcal{L}_{CE} = -\sum_{(x,y)} \log P(y|x; f_i)$ via stochastic gradient descent where (x, y) is the paired data, or using unsupervised learning techniques to obtain like unsupervised NMT [11] or unsupervised image translation [33]. Different methods can be used to obtain diverse agents $f_i, i \in \{1, 2, \cdots, N-1\}$, including using various random seeds to affect weight initialization and input order of the training samples, different model architectures, or training over different subsets. The same holds for $g_i, i \in \{1, 2, \cdots, N-1\}$. In the training process of f_0 and g_0, all these pretrained $2(N-1)$

models remain fixed, and are linearly aggregated with f_0, g_0 into two mixed models $F_{\boldsymbol{\alpha}}$ and $G_{\boldsymbol{\beta}}$. Specifically, given any $\alpha_i \geq 0$ and $\beta_i \geq 0$ for any $i \in \{0, 1, \cdots, N-1\}$,

$$\begin{aligned} F_{\boldsymbol{\alpha}} = \sum_{i=0}^{N-1} \alpha_i f_i,\ G_{\boldsymbol{\beta}} = \sum_{j=0}^{N-1} \beta_j g_j; \\ \text{s.t. } \sum_{i=0}^{N-1} \alpha_i = 1,\ \sum_{j=0}^{N-1} \beta_j = 1. \end{aligned} \tag{3.2}$$

For multi-agent dual learning, the duality feedback signal is built upon $F_{\boldsymbol{\alpha}}$ and $G_{\boldsymbol{\beta}}$. Following the basic framework of dual learning [24], for any $x \in \mathcal{X}$, all agents first cooperate to generate a $\hat{y} \in \mathcal{Y}$ by $\hat{y} = F_{\boldsymbol{\alpha}}(x)$, and then jointly reconstruct the $\hat{x} \in \mathcal{X}$ through $\hat{x} = G_{\boldsymbol{\beta}}(\hat{y})$. The reconstruction error between $y \in \mathcal{Y}$ and $\hat{y} = F_{\boldsymbol{\alpha}}(G_{\boldsymbol{\beta}}(y))$ is similarly constructed. The out-coming dual learning loss is defined as

$$\begin{aligned} \mathcal{L}_{\text{MADL}}(\mathcal{D}_x, \mathcal{D}_y; F_{\boldsymbol{\alpha}}, G_{\boldsymbol{\beta}}) = & \frac{1}{|\mathcal{D}_x|} \sum_{x \in \mathcal{D}_x} \Delta_x(x, G_{\boldsymbol{\beta}}(F_{\boldsymbol{\alpha}}(x))) \\ & + \frac{1}{|\mathcal{D}_y|} \sum_{y \in \mathcal{D}_y} \Delta_y(y, F_{\boldsymbol{\alpha}}(G_{\boldsymbol{\beta}}(y))). \end{aligned} \tag{3.3}$$

In this way, N agent pairs (prima-dual model pairs) are involved to jointly train and improve f_0 and g_0. When $N = 1$, Our algorithm degenerates to standard dual learning. No labeled information is required for building $\mathcal{L}_{\text{MADL}}$. We can apply it on either labeled data or unlabeled data.

Note that Equation (3.3) is the loss about duality. Other training objectives could also be included. For example, in NMT, if bilingual data is available, the cross entropy loss can be included to guide the training; in image-to-image translation, the GAN loss could also be included to enforce the generated images into the correct categories. In the next two sections, we will discuss how to adapt Equation (3.3) to different applications.

Discussion While there are existing works using multiple agents to boost the model performance, none of them has leveraged the duality. We use the $\mathcal{X} \to \mathcal{Y}$ mapping task to compare the previous work with our proposed framework:

- **Ensemble Learning**: Ensemble learning [54] is a straightforward way to combine multiple models during inference. To predict the label of $x \in \mathcal{X}$, all agents vote together and the final label would be $\arg\min_{y \in \mathcal{Y}} \sum_{i=0}^{N-1} \alpha_i \mathcal{L}(f_i(x), y)$, where $\mathcal{L}$ is the

loss function over space $\mathcal{X} \times \mathcal{Y}$. The α_i's can be simply set as $1/N$ or adaptively set according to the quality of each agent. There are several differences between ensemble learning and our work: First, ensemble learning does not use multiple agents in training as we do. Second, our multi-agent dual learning uses only one model f_0 in inference, which is more efficient than ensemble learning that uses multiple agents. Third, duality is not considered in conventional ensemble learning.

- **Knowledge Distillation**: Knowledge distillation with multiple agents [55, 56] is another line of work that combines multiple models. Knowledge distillation consists of two steps. First, all f_i's generate soft labels for $x \in \mathcal{X}$, e.g., $\hat{y} = \arg\min_{y \in \mathcal{Y}} \sum_{i=0}^{N-1} \alpha_i \mathcal{L}(f_i(x), y)$. Second, the generated pairs $(x, \hat{y})$'s are together used to train a new model. Each $(x, \hat{y})$ is regarded as labeled data without evaluating the quality of $\hat{y}$ or considering whether it is good enough for model training. In our proposed framework, we leverage the duality to build a feedback loop so as to evaluate the quality of generated pairs.

3.3 ADAPTATION TO NEURAL MACHINE TRANSLATION

In this section, we introduce how to adapt the general multi-agent dual learning (MADL) framework to Neural Machine Translation (NMT). Note that for NMT, $\mathcal{X}$ and $\mathcal{Y}$ stands for the collection of all possible sentences of two given languages, while $\mathcal{D}_x$ and $\mathcal{D}_y$ is the dataset. Denote the parameters of mapping f_0 and g_0 as θ_0^f and θ_0^g respectively. Following the common practice in NMT, the Δ_x and Δ_y is specified as the negative log-likelihood, that is, for any $x \in \mathcal{D}_x$ or $x \in \mathcal{X}$,

$$\begin{aligned}
&\Delta_x(x, G_{\boldsymbol{\beta}}(F_{\boldsymbol{\alpha}}(x))) \\
&= -\log \sum_{\hat{y} \in \mathcal{Y}} P(F_{\boldsymbol{\alpha}}(x) = \hat{y} | x; F_{\boldsymbol{\alpha}}, G_{\boldsymbol{\beta}}) P(G_{\boldsymbol{\beta}}(\hat{y}) = x | x, F_{\boldsymbol{\alpha}}(x) = \hat{y}; F_{\boldsymbol{\alpha}}, G_{\boldsymbol{\beta}}) \\
&= -\log \sum_{\hat{y} \in \mathcal{Y}} P(F_{\boldsymbol{\alpha}}(x) = \hat{y} | x; F_{\boldsymbol{\alpha}}) P(G_{\boldsymbol{\beta}}(\hat{y}) = x | \hat{y}; G_{\boldsymbol{\beta}}).
\end{aligned} \tag{3.4}$$

For ease of reference, we briefly denote Equation (3.4) as

$$\Delta_x(x, G_{\boldsymbol{\beta}}(F_{\boldsymbol{\alpha}}(x))) = -\log \sum_{\hat{y} \in \mathcal{Y}} P(\hat{y} | x; F_{\boldsymbol{\alpha}}) P(x | \hat{y}; G_{\boldsymbol{\beta}}). \tag{3.5}$$

Similarly, for any $y \in \mathcal{Y}$, we have

$$\Delta_y(y, F_{\boldsymbol{\alpha}}(G_{\boldsymbol{\beta}}(y))) = -\log \sum_{\hat{x}\in\mathcal{X}} P(\hat{x}|y; G_{\boldsymbol{\beta}}) P(y|\hat{x}; F_{\boldsymbol{\alpha}}). \tag{3.6}$$

Following Hassan et al. [30], we minimize the upper bounds of Δ_x and Δ_y for ease of optimization, which are respectively denoted as $\bar{\Delta}_x$ and $\bar{\Delta}_y$:

$$\begin{aligned}
\bar{\Delta}_x(x, G_{\boldsymbol{\beta}}(F_{\boldsymbol{\alpha}}(x))) &= -\sum_{\hat{y}\in\mathcal{Y}} P(\hat{y}|x; F_{\boldsymbol{\alpha}}) \log P(x|\hat{y}; G_{\boldsymbol{\beta}}) \geq \Delta_x(x, G_{\boldsymbol{\beta}}(F_{\boldsymbol{\alpha}}(x))); \\
\bar{\Delta}_y(y, F_{\boldsymbol{\alpha}}(G_{\boldsymbol{\beta}}(y))) &= -\sum_{\hat{x}\in\mathcal{X}} P(\hat{x}|y; G_{\boldsymbol{\beta}}) \log P(y|\hat{x}; F_{\boldsymbol{\alpha}}) \geq \Delta_y(y, F_{\boldsymbol{\alpha}}(G_{\boldsymbol{\beta}}(y))).
\end{aligned} \tag{3.7}$$

where two $\geq$'s hold due to Jensen's inequality. Then we turn to minimize

$$\begin{aligned}
\tilde{\mathcal{L}}_{\text{dual}}(\mathcal{D}_x, \mathcal{D}_y; F_{\boldsymbol{\alpha}}, G_{\boldsymbol{\beta}}) = &\frac{1}{|\mathcal{D}_x|} \sum_{x\in\mathcal{D}_x} \bar{\Delta}_x(x, G_{\boldsymbol{\beta}}(F_{\boldsymbol{\alpha}}(x))) \\
&+ \frac{1}{|\mathcal{D}_y|} \sum_{y\in\mathcal{D}_y} \bar{\Delta}_y(y, F_{\boldsymbol{\alpha}}(G_{\boldsymbol{\beta}}(y))).
\end{aligned} \tag{3.8}$$

Directly calculating the gradients for $\bar{\Delta}_x$ w.r.t θ_0^f and θ_0^g is difficult. First, $\mathcal{Y}$ is exponentially large since it represents all possible sentences of a language, which makes the computation intractable. Second, all f_i's and g_i's ($i \geq 1$) need to be loaded into the memory, although they will not be updated, which makes the computation inefficient. Similar issues exist for computing the gradients of $\bar{\Delta}_y$ w.r.t θ_0^f and θ_0^g.

We use importance sampling to address the above two difficulties. Define weight $\boldsymbol{\gamma}$ as $\boldsymbol{\gamma} = (0, \frac{1}{N-1}, \cdots, \frac{1}{N-1})$, thus F_γ and G_γ represent the combined model from all the pretrained agents without the target models f_0 and g_0 (see Equation (3.2)). Define

$$\begin{aligned}
\delta(x, \hat{y}) &= \left(\frac{P(\hat{y}|x; F_{\boldsymbol{\alpha}})}{P(\hat{y}|x; F_\gamma)}\right) \log P(x|\hat{y}; G_{\boldsymbol{\beta}}), \\
\phi(y, \hat{x}) &= \left(\frac{P(\hat{x}|y; G_{\boldsymbol{\beta}})}{P(\hat{x}|y; G_\gamma)}\right) \log P(y|\hat{x}; F_{\boldsymbol{\alpha}}).
\end{aligned} \tag{3.9}$$

Then the gradients for $\bar{\Delta}_x$ w.r.t θ_0^f is:

$$\frac{\partial \bar{\Delta}_x}{\partial \theta_0^f} = -\sum_{\hat{y}\in\mathcal{Y}} P(\hat{y}|x; F_\gamma) \frac{\partial \delta(x, \hat{y})}{\partial \theta_0^f} \tag{3.10}$$

and can be estimated by:

Algorithm 3.1: Algorithm for Multi-Agent Dual Learning.

1 **Input:** Data $\mathcal{D}_x$ and $\mathcal{D}_y$; Learning rate η; f_i and g_i $i \in \{0, 1, \cdots, N-1\}$; Mini-batch size K; Bilingual data $\mathcal{B}$ if possible;

2 Define $\gamma = (0, \frac{1}{N-1}, \cdots, \frac{1}{N-1})$ and $\alpha = \beta = (\frac{1}{N}, \frac{1}{N}, \cdots, \frac{1}{N})$;

3 Sample the datasets $\mathcal{D}_x$ and $\mathcal{D}_y$; obtain $\hat{\mathcal{D}}_y = \{\hat{y} \sim F_\gamma(x) | x \in \mathcal{D}_x\}$, $\hat{\mathcal{D}}_x = \{\hat{x} \sim G_\gamma(y) | y \in \mathcal{D}_y\}$;

4 Compute the probabilities $P(\hat{y}|x; F_\gamma)$ and $P(\hat{x}|y; G_\gamma)$;

5 **while** *not converged* **do**

6 Randomly sample two batches of $\mathcal{B}_{x\hat{y}} \subset \mathcal{D}_x \times \hat{\mathcal{D}}_y$ and $\mathcal{B}_{\hat{x}y} \subset \hat{\mathcal{D}}_x \times \mathcal{D}_y$, each of size K;

7 Following Equation (3.9) and Equation (3.11), calculate Grad_{f_0} and Grad_{g_0} as follows: $\text{Grad}_{f_0} \leftarrow -\frac{1}{|\mathcal{B}_{x\hat{y}}|} \sum_{(x,\hat{y}) \in \mathcal{B}_{x\hat{y}}} \frac{\partial \delta(x,\hat{y})}{\partial \theta_0^f} - \frac{1}{|\mathcal{B}_{\hat{x}y}|} \sum_{(\hat{x},y) \in \mathcal{B}_{\hat{x}y}} \frac{\partial \phi(y,\hat{x})}{\partial \theta_0^f}$ and $\text{Grad}_{g_0} \leftarrow -\frac{1}{|\mathcal{B}_{x\hat{y}}|} \sum_{(x,\hat{y}) \in \mathcal{B}_{x\hat{y}}} \frac{\partial \delta(x,\hat{y})}{\partial \theta_0^g} - \frac{1}{|\mathcal{B}_{\hat{x}y}|} \sum_{(\hat{x},y) \in \mathcal{B}_{\hat{x}y}} \frac{\partial \phi(y,\hat{x})}{\partial \theta_0^g}$;

8 If bilingual data is available, sample a batch $\mathcal{B}_{xy} \subset \mathcal{B}$ of size K, calculate $\text{Grad}_{f_0} \leftarrow \text{Grad}_{f_0} - \frac{1}{K} \nabla_{\theta_0^f} \sum_{(x,y) \in \mathcal{B}_{xy}} \log P(y|x; f_0)$ and $\text{Grad}_{g_0} \leftarrow \text{Grad}_{g_0} - \frac{1}{K} \nabla_{\theta_0^g} \sum_{(x,y) \in \mathcal{B}_{xy}} \log P(x|y; g_0)$;

9 Update the parameters: $\theta_0^f \leftarrow \theta_0^f - \eta \, \text{Grad}_{f_0}$, $\theta_0^g \leftarrow \theta_0^g - \eta \, \text{Grad}_{g_0}$;

10 **end**

1. Given an $x \in \mathcal{D}_x$ and $y \in \mathcal{D}_y$, sample a $\hat{y}$ according to the distribution $P(\cdot|x; F_\gamma)$; sample an $\hat{x}$ according to the distribution $P(\cdot|y; G_\gamma)$;

2. Estimate the gradients of $\bar{\Delta}_x$ w.r.t θ_0^f as follows (and similarly for θ_0^g without loss of generality):

$$\frac{\partial \bar{\Delta}_x}{\partial \theta_0^f} = \mathbb{E}_{\hat{y} \sim P(\cdot|x;F_\gamma)} \left[-\frac{\partial \delta(x,\hat{y})}{\partial \theta_0^f} \right] \approx -\frac{\partial \delta(x,\hat{y})}{\partial \theta_0^f}. \tag{3.11}$$

Our method is summarized in Algorithm 3.1. We do the offline sampling by sampling $\hat{x}$'s and $\hat{y}$'s in advance (step 3); and computing the probabilities related to f_i's and g_i's $i \geq 1$ in advance (step 4). When bilingual data is available, denoted as $\mathcal{B} = \{(x_k, y_k)\}_{k=1}^M$ where M is the number of training data, we can also apply the negative log-likelihood loss on bilingual data (step 8).

3.4 EXPERIMENTS

We conduct extensive experiments on various neural machine translation tasks, including supervised translation with only bilingual training data, semi-supervised translation with additional monolingual data, and unsupervised translation without any bilingual data.

3.4.1 Experiment Settings

Dataset We evaluate MADL on multiple benchmark NMT datasets, including IWSLT 2014 English↔German translation, WMT 2014 English↔German translation, and WMT 2016 unsupervised English↔German translation. English and German is denoted as "En" and "De" respectively for ease of reference.

- *IWSLT 2014 En↔De*: following Edunov et al. [57], we lowercase all the sentences, and split them into training/validation/test set with $153k/7k/7k$ sentences respectively.
- *WMT 2014 En↔De*: we choose WMT 2014 training set and filter out $4.5M$ sentences pairs following Gehring et al. [6] and Vaswani et al. [7]. We concatenate newstest 2012 and newstest 2013 as the validation set and use newstest 2014 as the test set. We also select $8M$ English and $8M$ German monolingual sentences from newscrawl 2013 as monolingual dataset.
- *Unsupervised En↔De*: following Lample et al. [11], we choose 50M monolingual English and German sentences and use newstest 2016 as the test set for a fair comparison with the previous work. We preprocess the words into word-pieces in the same way as Wu et al. [5].

Model Architecture We use Transformer [7] as the basic model structure. For IWSLT En↔De, we use the *transformer_small* configuration with 4 and 8 blocks to verify the generality of our model with different structures (a 4-block setting refer to transformer with 4 encoder blocks and 4 decoder blocks), where the word embedding dimension, hidden state dimension and non-linear layer dimension is set to be 256, 256 and 1024 respectively. For the WMT task, following Vaswani et al. [7], we use the *transformer_big* setting with 6 blocks, where the word embedding dimension, hidden state dimension and non-linear layer dimension is 1024, 1024 and 4096 respectively. For the unsupervised NMT task, we choose *transformer_base* setting following Lample et al. [11], where the above three dimensions are 512, 512 and 2048. The dropout rates for the three settings are 0.2, 0.1 and 0.1 respectively.

Optimization and Evaluation We choose Adam [58] to optimize the network. The η in Algorithm 3.1 is set as 2×10^{-4}. The learning rate decay rule and the two β's of Adam are the same as Vaswani et al. [7]. For the three tasks, we use one, eight and four M40 GPUs to train those networks for three, five and six days respectively. Beam search is applied to generate translations for all the models, where the beam sizes for the three tasks are 6, 4

Table 3.1: BLEU scores on IWSLT De↔En translation. "KD", "BT" and "Dual" stands for knowledge distillation, back translation and dual learning respectively.

	Standard	KD-1	KD-5	BT-1	BT-5	Dual-1	Dual-5
De→En (4B)	33.42	33.89	34.20	33.71	33.61	34.25	**34.70**
En→De (4B)	27.89	28.45	28.65	28.35	28.22	28.63	**28.99**
De→En (8B)	34.01	34.36	34.85	33.87	33.77	34.57	**35.44**
En→De (8B)	27.95	28.74	29.18	28.28	28.25	29.07	**29.52**

and 4 respectively. The evaluation metric is BLEU score [59], which is the geometric mean of n-gram precisions, $n \in \{1, 2, 3, 4\}$.

Baselines We implement three types of important baselines, including back translation [3, 29], knowledge distillation [56] and the two-agent dual learning [24]. All the implementations are based on the official tensor2tensor release[3]. Take the $f : \mathcal{X} \mapsto \mathcal{Y}$ translation as an example:

- *Back Translation (BT)*: a reversed translation model $g : \mathcal{Y} \mapsto \mathcal{X}$ is first trained on the original bilingual dataset, then the dataset $\{(g(y), y)|y \in \mathcal{D}_y\}$ is constructed, which is then concatenated with the original dataset and train another model f.
- *Knowledge Distillation (KD)*: we pre-train a teacher model $f_T : \mathcal{X} \to \mathcal{Y}$, and use it to generate the corresponding dataset $\{(x, f_T(x))|x \in \mathcal{D}_x\}$, which is then mixed with the ground truth data and used to train a new model[4]
- *Two-agent dual learning*: we use f together with g to build the feedback signal to regularize the training (see Equation (3.1)), which we denote as "Dual-1".

For the first two baselines BT and KD, we generate sentences using both a single model and multiple models. Our multi-agent models are denoted as "Dual-i" ($i > 1$).

3.4.2 Results on IWSLT Dataset

We regard the English and German sentences in the IWSLT14 bilingual corpus as $\mathcal{D}_x$ and $\mathcal{D}_y$ in Algorithm 3.1. We use 4-block networks (i.e. F_γ in Algorithm 3.1) to generate translations, which are then used to train both 4-block (4B) and 8-block (8B) networks. We

[3] `https://github.com/tensorflow/tensor2tensor`

[4] We have tried both mixing with ground truth and not and found that mixing is better. On IWSLT 2014 De→En, with/without mixing can achieve 33.89/33.14 BLEU scores.

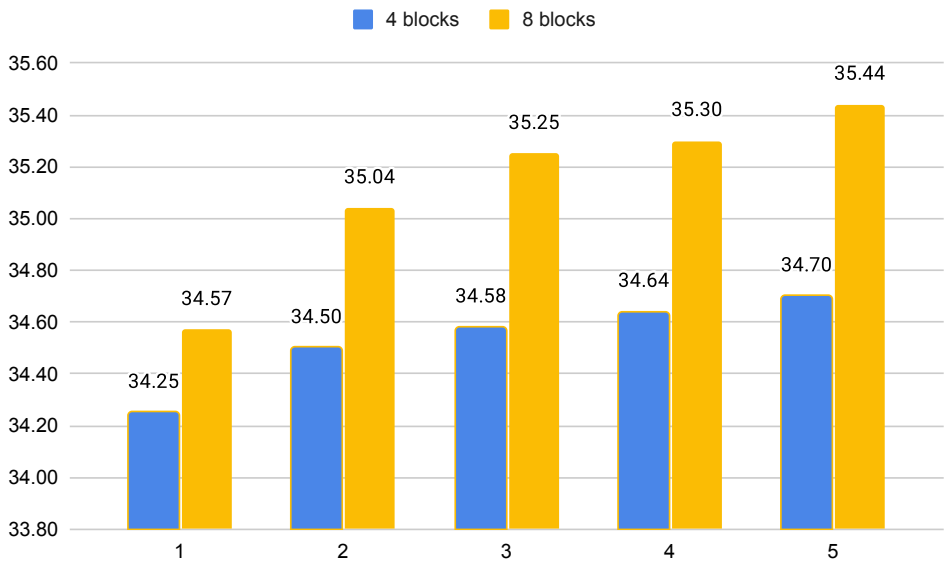


Figure 3.1: BLEU scores on IWSLT De→En w.r.t the number of agents pairs.

compare dual learning (Dual-i) with the standard baseline (Standard), knowledge distillation (KD-i) and back translation (BT-i) with different number of primal-dual model pairs $i \in \{1, 5\}$. Specifically, Dual-1 is the two-agent dual learning baseline, and Dual-5 is our multi-agent model with 4 additional pairs of agents.

The results of IWSLT En↔De are presented in Table 3.1. We can observe that:

1. Dual learning has brought significant improvement over all the baselines (Standard, KD and BT) in both single-agent and multi-agent settings, demonstrating the effectiveness of dual learning.

2. Involving multiple agents into the learning system leads to better performances, which shows the importance of additional feedback signals from other agents. In particular, Dual-5 outperforms Dual-1 by around 0.5 BLEU in 4B setting and around 0.8 BLEU in 8B setting, which proves that dual learning can benefit from cooperating with more agents and demonstrates the effectiveness of our proposed algorithm.

3. Particularly, dual learning with $N = 5$ agent pairs under 8B setting sets a new record of 35.44 BLEU in De→En and 29.52 BLEU in En→De translations.

Study on different number of agents We study the performances of our proposed algorithm with respect to different numbers of agent pairs, i.e., N and explore the optimal value of N. We enumerate within $N = \{1, \cdots, 5\}$ and did not try larger N values due to the

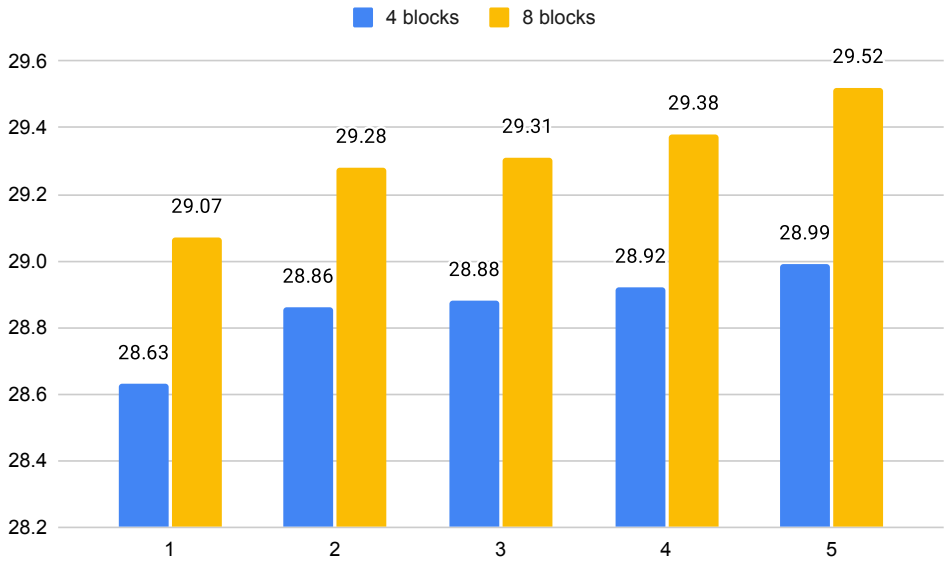


Figure 3.2: BLEU scores on IWSLT En→De w.r.t the number of agents pairs.

limitation of computational resources. As we can see from Figure 3.1 and Figure 3.2, more agents can bring better performance for all settings. While it appears that the performance becomes better with more pair of agents involved into the learning system, the gain becomes more and more marginal (e.g., the maximum difference for $N = 4$ and $N = 5$ across all settings is around 0.1 BLEU). Furthermore, larger N leads to higher computational costs such as GPU resources. Thus, it is the most practical to leverage $N = 3$ agent pairs, where we can benefit from the substantial gain over baselines without too much computational costs. We employ $N = 3$ for the rest of our experiments.

Study on diversity of agents We further study how the diversity among agents influences the performances of our algorithm on the IWSLT De↔En dataset. We compare three group of agents:

- Agents with independent runs (*Independent Run*): We use the same agents as Section 3.4.2, which have the same model architecture and hyperparameters, and are trained through independent runs with different random seeds.

- Agents with different architectures (*Different Arch*): The agents have different hidden dimensions (256, 512) with different numbers of heads (2, 4). They are also trained through independent runs with different random seeds. Intuitively, this group of agents with different architectures could be more diverse than the first group.

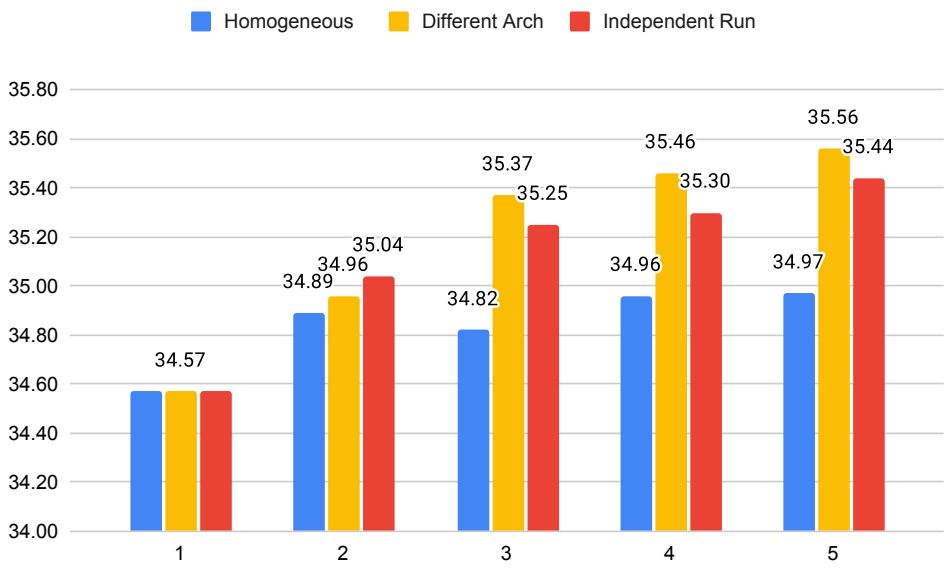


Figure 3.3: BLEU scores on IWSLT De→En with different pools of agents.

- Agents from a same run ("*Homogeneous*"): In comparison to the first two groups where different methods (i.e., different random seeds, different model architecture) are used to encourage diversity among agents, in this group, we use model checkpoints at different (but very close) iterations from a same run. At the late stage of training, the learning rate becomes smaller. Thus the diversity among agents would naturally be very poor.

We train 8-block (8B) networks with $N = \{1, ..., 5\}$ pairs of agents from the above three groups respectively. The same training and inference procedures as Section 3.4.2 are applied here. As is shown from results in Figure 3.3 and Figure 3.4, we have the following observations and conclusions:

1. Diversity among agents plays an important role in our method. The *Homogeneous* agents with poor diversity has the worst performances among the three groups. That is, introducing more homogeneous agents brings very marginal gain.

2. A more diverse group of agents can potentially bring further improvements, yet diversity is not the key focus of our algorithm. *Independent Run* achieves comparable performances with *Different Arch*, suggesting that our framework can achieve substantial improvements with a reasonable level of diversity among agents.

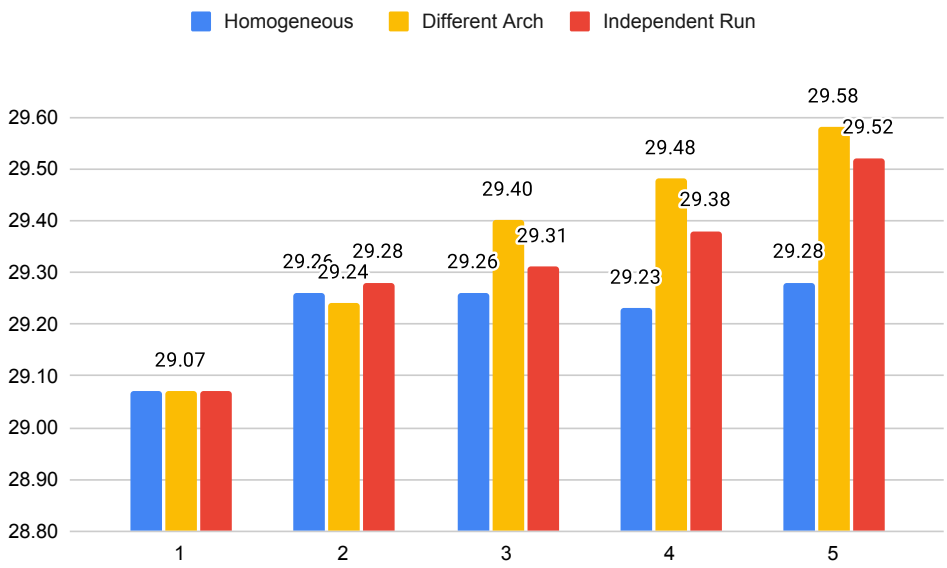


Figure 3.4: BLEU scores on IWSLT En→De with different pools of agents.

Table 3.2: BLEU scores on IWSLT {Es, Ru, He} ↔ En translations.

	Es→En	En→Es	Ru→En	En→Ru	He→En	En→He
Standard	40.50	37.95	18.94	16.06	33.25	22.20
Dual-1	42.14	38.09	21.14	16.31	35.42	22.75
Dual-3	42.41	38.62	21.56	16.71	36.08	23.80

Study on different language pairs We carry out another three groups of IWSLT14 experiments, English↔Spanish (Es), English↔Russian (Ru) and English ↔Hebrew (He) translations to verify the generality of our framework on different language pairs. As is illustrated in Table 3.2, we achieve consistent improvements across different translation tasks (for all languages and all translation directions), further demonstrating the effectiveness and robustness of our algorithm.

3.4.3 Results on WMT Dataset

We work with two settings on WMT 2014 En↔De task: (1) only bilingual data is available; and (2) additional monolingual data is provided. The results are summarized in Table 3.3. As we can see, the proposed algorithm achieves the best performances for all settings, and we observe that:

1. With WMT 2014 bilingual data only, the proposed multi-agent dual learning (Dual-3)

Table 3.3: BLEU scores on WMT14 En↔De translation. "Bitext" and "Mono" respectively represents using bilingual data only and using the mix of bilingual&monolingual data.

	Standard	KD-1	KD-3	BT-1	BT-3	Dual-1	Dual-3
En→De (Bitext)	28.4	29.17	29.28	28.26	27.79	29.44	**30.05**
De→En (Bitext)	32.15	32.43	31.38	32.41	32.68	32.99	**33.32**
En→De (Mono)	28.4	29.20	29.36	29.42	29.68	29.93	**31.03**
De→En (Mono)	32.15	32.28	32.43	34.58	34.77	34.98	**35.64**

achieves the state-of-the-art results to our knowledge, with 30.05 BLEU for En→De and 33.32 BLEU for De→En[5].

2. Compared with the traditional two-agent dual learning in [24], we can achieve 0.61 BLEU score improvement on En→De and 0.33 on De→En with bilingual data only. For the setting of using monolingual data, the improvements are 1.10 and 0.66 respectively.

3. Although KD and BT can bring improvements under different settings, the results are not as consistent as dual learning. For example, the performance of BT is not as good as KD under bilingual setting, but is better than KD under monolingual setting. We conjecture the reason is that with feedback signals to a generated sentence in the dual learning framework, the dataset is not only enlarged through sampling, but also guaranteed with good quality of generation.

Study on generality of the algorithm We apply our method to the state-of-the-art systems on WMT 2016, 2017 and 2018 English-to-German translation challenge to verify the generality of our method. The training data, $F_\gamma(\cdot)$ and $G_\gamma(\cdot)$ are the same as those used in Section 3.4.3. We work with a degenerated case where g_0 (i.e., the De→En model) is pre-trained before optimizing f_0 and subsequently hold constant. The f_0 is warm started by the state-of-the-art models that are well trained and released with tremendous resources[6]. We degenerate our algorithm to optimize the single direction due to the limitation that only En→De model checkpoints are released.

We train the model on 8 M40 GPUs for 2 days with PyTorch implementation of our algorithm based on FairSeq Toolkit[7] . The batch size is 4096 tokens. We use Adam optimizer and the learning rate is automatically calculated by the system based on the reloaded

[5]Although Edunov et al. [29] provides a higher BLEU score, they use WMT'18 dataset, which is different from ours and currently not widely used for WMT14 En↔De tasks in NMT literature

[6]`https://dl.fbaipublicfiles.com/fairseq/models/wmt18.en-de.ensemble.tar.bz2`

[7]`https://github.com/pytorch/fairseq`

Table 3.4: BLEU scores on WMT {16, 17, 18} En→De. The single models are independently trained for 6 times, with mean and standard derivation values reported; the ensemble models are the ensemble results of all different runs.

	2016	2017	2018
Google Translator	38.03	31.41	47.67
FAIR (Single)	37.04 ± 0.16	31.86 ± 0.21	44.63 ± 0.12
MADL (Single)	$\mathbf{40.68 \pm 0.11}$	$\mathbf{33.47 \pm 0.16}$	$\mathbf{48.89 \pm 0.13}$
MS-Marian (Ensemble)	39.6	31.9	48.3
FAIR (Ensemble)	37.99	32.80	46.05
MADL (Ensemble)	**41.23**	**34.01**	**49.61**

checkpoint. The dropout ratio is 0.3. We use beam search with beam size 5 to generate candidates. We adopt the commercial system Google Translator (crawled on Oct 16, 2018), the champion system of WMT18 En→De: MS-Marian [60], and FAIR's single and ensemble models [29] as our baselines. The models are evaluated by sacreBLEU[8].

As can be seen from Table 3.4, our method is capable of further improving the strong model by a large margin. Both single and ensemble models with our algorithm outperform the previous best systems and set state-of-the-art results on these tasks, demonstrating the effectiveness of our method.

3.4.4 Results on Unsupervised NMT

Unsupervised NMT is studied recently to learn two translation models without bilingual data. Similar to the standard NMT model with bilingual training data, an unsupervised NMT model is also based on the encoder-decoder architecture [11]. The source sentences and target sentences are mapped into a same vocabulary using BPE techniques. The shared embedding is pretrained with fastText [12] to get good initial values. The proposed model can handle both source-to-target translation and target-to-source translation [13].

The training loss usually consists of two parts, a language model loss and a back-translation loss. Let $P_{\mathcal{X}\to\mathcal{Y}}$ denote the translation from space $\mathcal{X}$ to space $\mathcal{Y}$, and so for the other similar notations.

- *Language model loss*: the language model loss is implemented by a denoising autoen-

[8]sig=BLEU+case.mixed+lang.en-de+numrefs.1+smooth.exp+test.wmt{16}+tok.13a+version.1.2.11

Table 3.5: BLEU scores on WMT 2016 unsupvised NMT En↔De translation.

	Standard	KD-1	KD-3	BT-1	BT-3	Dual-1	Dual-3
En→De	17.52	17.33	18.27	17.50	18.15	18.01	**19.26**
De→En	22.12	22.10	22.71	22.51	22.91	22.17	**23.85**

coder. Mathematically,

$$L_{LM} = \mathbb{E}_{x\sim\mathcal{X}}[-\log P_{\mathcal{X}\to\mathcal{X}}(x|C(x))] + \mathbb{E}_{y\sim\mathcal{Y}}[-\log P_{\mathcal{Y}\to\mathcal{Y}}(y|C(y))], \tag{3.12}$$

where $C(\cdot)$ is a noise model that randomly drops or swaps words.

- *Back translation loss*, implemented by back-translating the monolingual data and feeding into the reversed models. Mathematically,

$$L_{BT} = \mathbb{E}_{x\sim\mathcal{X}}[-\log P_{\mathcal{Y}\to\mathcal{X}}(x|\hat{y}(x))] + \mathbb{E}_{y\sim\mathcal{Y}}[-\log P_{\mathcal{X}\to\mathcal{Y}}(y|\hat{x}(y))], \tag{3.13}$$

in which $\hat{y}(x) = \arg\max_{u\in\mathcal{Y}} P_{\mathcal{X}\to\mathcal{Y}}(u|x)$ and $\hat{x}(y) = \arg\max_{v\in\mathcal{X}} P_{\mathcal{Y}\to\mathcal{X}}(v|y)$. Note that the four $P_{\ldots}$'s are implemented in a single encoder-decoder based model, where each translation task has a different "task embedding", i.e., a learnable vector indicating the translating directions.

We pre-train two unsupervised NMT models with different initialization, use them to translate the $50M$ monolingual sentences, and apply KD, BT and dual learning algorithms.

As is shown in Table 3.5, with one pair of agents, KD-1 and BT-1 obtains almost the same results as the standard baseline, while Dual-1 achieves slightly better results on En→De translation. When we increase the number of agents, there are substantial improvements for KD-3, BT-3 and Dual-3. Especially, multi-agent dual learning (Dual-3) achieves 19.26 and 23.85 BLEU scores, setting new records for unsupervised NMT with pure NMT models.

3.5 APPLICATION TO WMT2019 NEWS TRANSLATION TASK

To further demonstrate the effectiveness of the proposed approach, we have MADL integrated into the translation systems submitted to the WMT2019 News Translation shared task [9] in English↔German and German↔French translation. All the four systems rank first place in the automatic evaluation with cased SacreBLEU.

[9] `https://www.statmt.org/wmt19/translation-task.html`

3.5.1 Experiment Setups

Dataset We concatenate "Europarl v9", "News Commentary v14", "Common Crawl corpus" and "Document-split Rapid corpus" as the basic bilingual dataset (denoted as $\bar{\mathcal{B}}_0$). We sample 20M data from "Paracrawl" and concatenated them together with $\bar{\mathcal{B}}_0$. The concatenated corpus is denoted as $\bar{\mathcal{B}}_1$. We clean $\bar{\mathcal{B}}_1$ by normalizing the sentences, removing non-printable characters, and tokenization. We share a vocabulary for the two languages and apply BPE for word segmentation with 35,000 merge operations. (We tried different BPE merge operations but found no significant differences.) For monolingual data, we use 120M English sentences (denoted as Men) and 120M German sentences (denoted as Mde) from Newscrawl, and preprocess them in the same way as bilingual data. We use newstest 2016 and the validation set and newstest 2018 as the test set.

Model Configuration We use the PyTorch implementation of Transformer. We choose the Transformer big setting, in which both the encoder and decoder are of six layers. The dropout rate is fixed as 0.2. We set the batch size as 4096 and the parameter "update-freq" as 16. We apply Adam optimizer with learning rate 5×10^{-4}.

3.5.2 Training Pipeline

The pipeline consists of three steps:

- Pre-train two English→German translation models (denoted as $\bar{f}_1$ and $\bar{f}_2$) and two German→English translation models (denoted as $\bar{g}_1$ and $\bar{g}_2$) on $\bar{\mathcal{B}}_1$; pre-train another English→German (denoted as $\bar{f}_3$) and German→English (denoted as $\bar{g}_3$) on $\bar{\mathcal{B}}_0$.

- Apply back translation: We back translate $\bar{\mathcal{M}}_{en}$ and $\bar{\mathcal{M}}_{de}$ using $\bar{f}_3$ and $\bar{g}_3$ with beam search, add noise to the translated sentences, merge the synthetic data with $\bar{\mathcal{B}}_1$, and train one English→German model f_0 and one German→English model g_0 for seven days on eight V100 GPUs.

- Apply MADL to f_0 and g_0. That is, the $F_{\boldsymbol{\alpha}}$ is specified as the combination of f_0, $\bar{f}_1$ and $\bar{f}_2$ with equal weights; and $G_{\boldsymbol{\beta}}$ consists of g_0, $\bar{g}_1$ and $\bar{g}_2$. During training, we will only update f_0 and g_0. To speed up training, we randomly select 20M monolingual English and German sentences from $\bar{\mathcal{M}}_{en}$ and $\bar{\mathcal{M}}_{de}$ respectively instead of using all monolingual sentences. The eventual output models are denoted as f_1 and g_1 respectively. This step takes 3 days on four P40 GPUs.

Table 3.6: Results of English↔German by sacreBLEU on the dev sets, compared with Transformer baseline, Back Translation (BT) and MADL (submitted system).

	En→De		De→En	
	newstest16	newstest18	newstest16	newstest18
Baseline	37.4	45.6	41.9	44.9
BT	39.2	47.4	45.0	47.1
MADL	41.9	50.4	47.4	49.1

Table 3.7: Submitted systems on WMT 2019 shared task (newstest2019)

	En→De	De→En	De→Fr	Fr→De
MADL	44.9	42.8	37.3	35.0
Others' Best	43.9	40.8	35.6	32.7

3.5.3 Results

The results are summarized in Table 3.6, which are evaluated by cased sacreBLEU. The baseline is the average accuracy of models using only bitext, i.e., $\bar{f}_1$ and $\bar{f}_2$ for English→German translation and $\bar{g}_1$ and $\bar{g}_2$ for German→English, and BT is the accuracy of the model after back-translation training. As can be seen, back translation improves accuracy. For example, back-translation boosts the BLEU score from 45.6 to 47.4 on news18 English→German translation, which is 1.8 point improvement. MADL further boosts BLEU to 50.4, obtaining another 3 BLEU improvement, demonstrating the effectiveness of our method.

For the final submission [2], we accumulate many translation models (trained using bitext, back translation, and MADL, with different random seeds) and do knowledge distillation on the source sentences from WMT14 to WMT19 test sets. Table 3.7 shows the performance of our system comparing to the best systems from the other participators. As can be seen from the table, we ranked in the first place in the four translation tasks, and outperformance the other best systems with large margin.

3.6 EXTENTION TO IMAGE TRANSLATION

As is introduced in Section 3.2, the *Multi-Agent Dual Learning (MADL)* framework is a general framework that can be easily adapted to various tasks with structural duality beyond NMT. In this section, we adapt MADL to the image translation tasks to further demonstrate the generality of the proposed framework. Different from [61] which also uses multiple agents (several columns of deep neural networks), we leverage the duality of image translation tasks

and only use one model during inference. We follow the setting of CycleGAN [33], the most popular implementation of image translation that combines the ideas of GAN [62] and dual learning.

3.6.1 Adaption

Following the common practice in image translation [33], for each term in Equation (3.3), the Δ_x and Δ_y are specified as the $L1$ norm of pixel-level difference between two images. We set $\alpha = \beta = (\frac{1}{N}, \frac{1}{N}, \cdots, \frac{1}{N})$. Since there is few literature focusing on generating consensus images using multiple models, we switch Equation (3.3) to a simpler form:

$$\mathcal{L}_{\text{dual}}(\mathcal{D}_x, \mathcal{D}_y; F_{\boldsymbol{\alpha}}, G_{\boldsymbol{\beta}}) = \frac{1}{|\mathcal{D}_x|} \sum_{x \in \mathcal{D}_x} \|x - G_{\boldsymbol{\beta}}(f_0(x))\|_1 + \frac{1}{|\mathcal{D}_y|} \sum_{y \in \mathcal{D}_y} \|y - F_{\boldsymbol{\alpha}}(g_0(y))\|_1, \quad (3.14)$$

where $\mathcal{D}_x$ and $\mathcal{D}_y$ are the datasets of domain $\mathcal{X}$ and domain $\mathcal{Y}$. The remaining parts of our model, including the model components and the objective functions, are exactly the same as the standard CycleGAN.

There are two discriminators $d_{\mathcal{X}} : \mathcal{X} \mapsto [0, 1]$ and $d_{\mathcal{Y}} : \mathcal{Y} \mapsto [0, 1]$ used to differentiate the generated images from natural images. The outputs of $d_{\mathcal{X}}$ and $d_{\mathcal{Y}}$ represent the probability that the input image is a natural one. The GAN loss [62] is defined as

$$\begin{aligned} \mathcal{L}_{\text{GAN}} = & \frac{1}{|\mathcal{D}_x|} \sum_{x \in \mathcal{X}} [\log d_{\mathcal{X}}(x) + \log \big(1 - d_{\mathcal{Y}}(f_0(x))\big)] \\ & + \frac{1}{|\mathcal{D}_y|} \sum_{y \in \mathcal{Y}} [\log d_{\mathcal{Y}}(y) + \log \big(1 - d_{\mathcal{X}}(g_0(y))\big)]. \end{aligned} \quad (3.15)$$

During the optimization process, the two discriminators will try to maximize ℓ_{GAN}, while the f_0 and g_0 will minimize $\ell_{\text{dual}}(\mathcal{D}_x, \mathcal{D}_y; F_{\boldsymbol{\alpha}}, G_{\boldsymbol{\beta}}) + \lambda \ell_{\text{GAN}}$ in cooperation with other provided agents, where λ is a hyper-parameter to be tuned. In the experiments, we follow Zhu et al. [33] to set λ as 10.

3.6.2 Settings

We use the same datasets, model configuration and optimization algorithm as [33]. We implement our method based on CycleGAN[10]. We choose two tasks, the photo↔painting

[10] `https://github.com/junyanz/pytorch-CycleGAN-and-pix2pix`

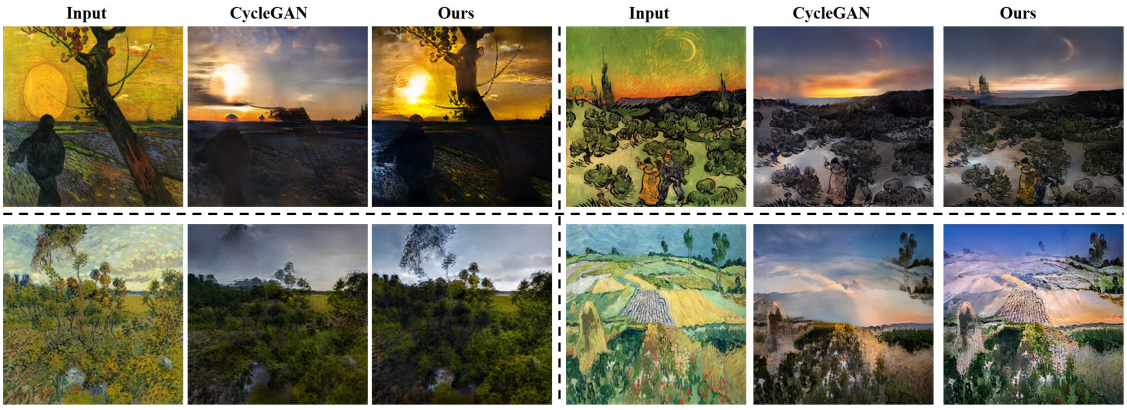


Figure 3.5: Four groups of experiments on painting→photo translations.

translations where the art styles include Monet, Van Gogh, Ukiyo-e, Cezanne, and cityscapes label↔photo translation. All the images are cropped to 256×256.

We follow [33] to configure the model architectures for all tasks on image-to-image translation. For the generator, the network contains 2 stride-2 convolutions, 9 residual blocks, and 2 convolutional layers with stride $\frac{1}{2}$, (i.e., the transposed convolutional layers). Instance normalization is applied to the network. For the discriminator, the 70×70 PatchGANs [63], which aim to classify whether 70×70 overlapping image patches are real or fake. Such a patch-level discriminator architecture has fewer parameters than a full-image discriminator and can work on arbitrarily-sized images in a fully convolutional fashion.

Apart from qualitative study, we further present the following evaluation metrics to quantitatively evaluate the model performances:

- **FID Evaluation**: Fréchet Inception Distance (FID) [64] is verified to be a reasonable metric [65] to evaluate the quality of painting→photo translations. FID captures the distance between generated images and the real ones, and a smaller FID score indicates the better model quality.

- **FCN Evaluation** FCN-scores are used in [33] and [63] to quantitatively evaluate Cityscape photo to semantic label translation. A pre-trained fully convolutional network [66], briefly, FCN, is used to convert the photo to the corresponding semantic label, and then calculate the per-pixel accuracy, per-class accuracy and mean class Intersection-Over-Union(Class IOU) [67] against the ground truth labels.

Table 3.8: FID scores on painting→photos translation.

	Cezanne	Monet	Ukiyo-e	Vangogh
CycleGAN	189.40	134.25	197.30	91.19
Ours	173.92	129.86	184.69	89.32

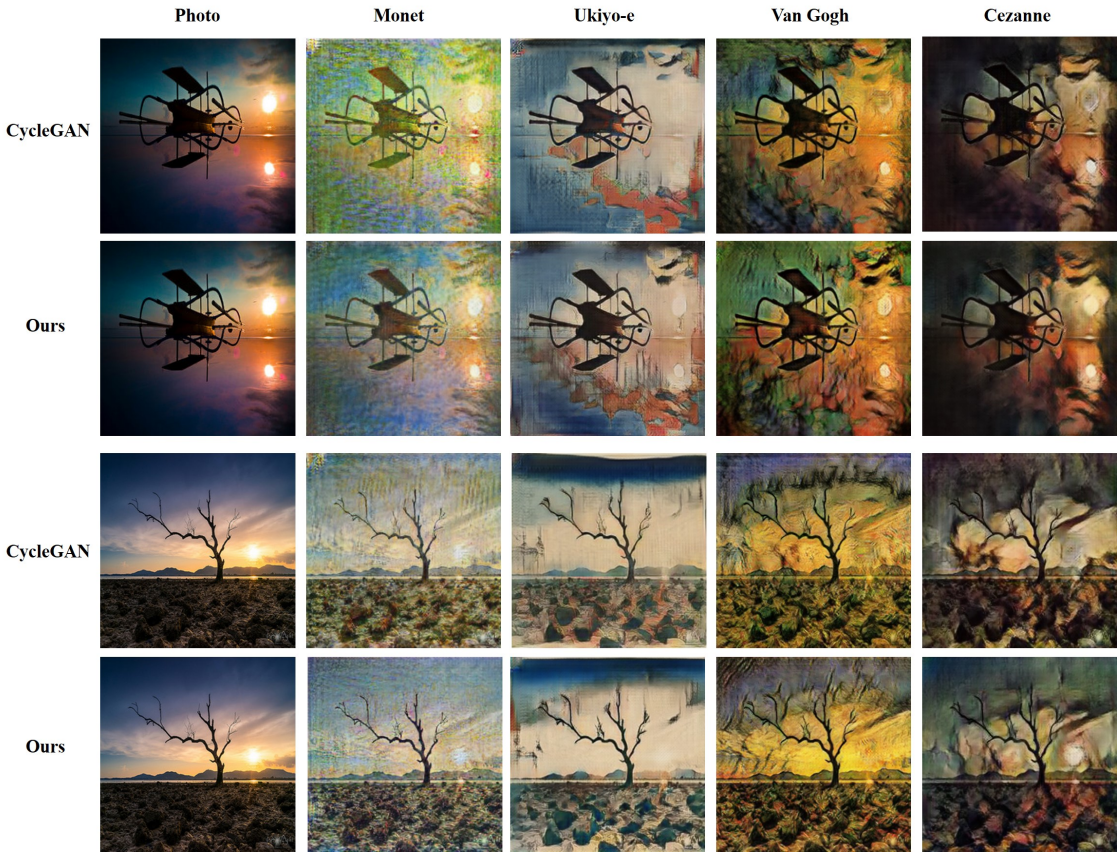


Figure 3.6: Two groups of experiments on photo→painting translations.

3.6.3 Results of Painting-to-Photo Translation

The task of painting→photo translation is to translate an input painting with specific art style to the original photo captured by a camera. We present case studies on painting→photo translations in Figure 3.5. Indeed, there are several cases that our algorithm is clearly better than CycleGAN: For the images in the top-left corner, CycleGAN fails to recover the tree while our algorithm succeeds, with many details correctly included like the branches and leaves. In the bottom right corner, the texture of the ground is kept by our results, which is missing from the images generated by CycleGAN.

Similarly, we present results on photo→painting translation in Figure 3.6. Both standard CycleGAN and our algorithm successfully translates the inputs to the corresponding target

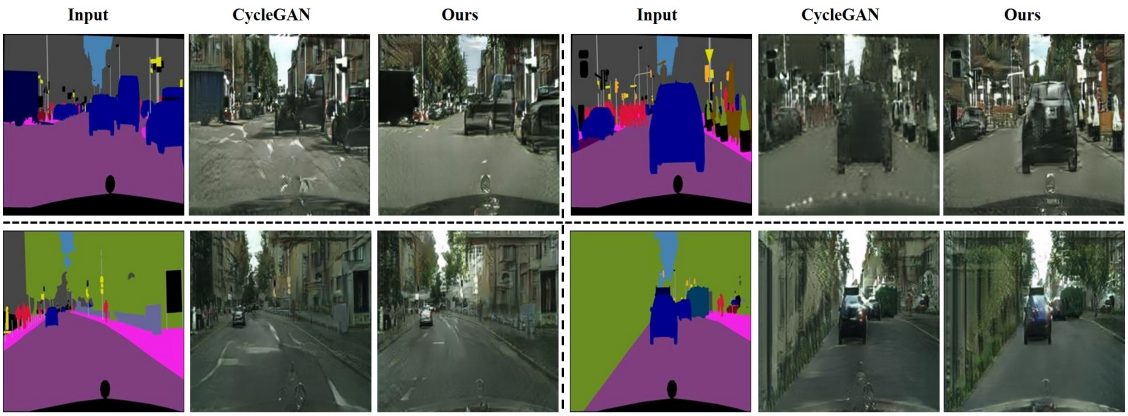


Figure 3.7: Cityscapes labels → photos translation.

Table 3.9: FCN-scores on cityscapes labels→photos translation.

	Per-pixel Acc.	Per-class Acc.	Class IOU
CycleGAN	0.52	0.17	0.11
Ours	0.57	0.19	0.14

domains. Our model outperforms the baseline in two aspects: 1) multi-agent dual learning can generate clearer images with less block artifact (the Monet column and Ukiyo-e column); and 2) the images generated by our algorithm are faithful to the original images in semantics. For example, the sun is missing from CycleGAN generated images in the top Ukiyo-e column and the bottom Cezanne column, while our algorithm is able to keep the sun.

3.6.4 Results of Cityscapes Label-to-Photo Translation

We present case studies as well as quantitative evaluation for label→photo translation on cityscapes images in Figure 3.7 and Table 3.8 respectively. As can be seen from Figure 3.7, both CycleGAN and our method can transfer the input label to a matching natural scene, while results of ours are clearer with less noise. For example, in the top left corner, our method can generate flatten roads, clearer cars and buildings and the baseline model fails. In Table 3.8, we present quantitative evaluation with "FCN-scores" following [63]. Our method achieves better results on the three evaluation metrics.

The example cases and quantitative evaluation with FCN-scores of photo→labels on cityscapes images are shown in Figure 3.8 and Table 3.10[11] respectively. We can observe

[11]The baseline performance is different from Zhu et al. [33] since it is unclear how the RGB based labels is converted to integer labels without the official evaluation code released. This is a common issue at `https://github.com/phillipi/pix2pix/issues/115`.

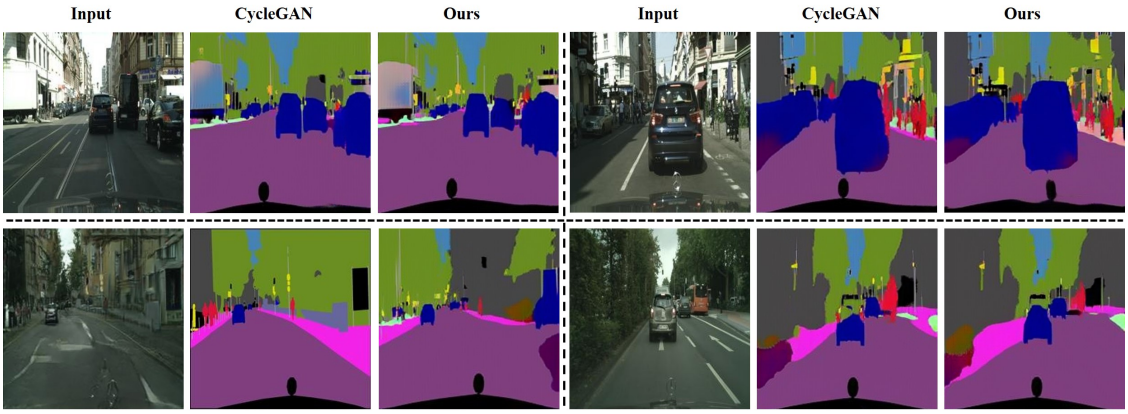


Figure 3.8: Cityscapes photo → labels translation

Table 3.10: Classification performances of Cityscapes photo → labels translation.

	Per-pixel Acc.	Per-class Acc.	Class IOU
Standard	0.56	0.19	0.15
Dual-3	0.57	0.20	0.15

improvements but the gap is not significant.

3.7 SUMMARY

In this chapter, we introduce a new framework, multi-agent dual learning, in which multiple primal models and dual models are involved in the learning process. We empirically verified the effectiveness of the proposed framework on multiple machine translation tasks and image translation tasks.

Extending the multi-agent dual learning framework to more applications would be an interesting direction for future research. For example, we can apply our framework to improve the recent non-autoregressive machine translation, which is known to have high inference efficiency yet poor translation quality [68, 69, 70]. It is also worth studying how to apply the framework to other dual learning paradigms, including dual supervised learning [1] and dual transfer learning [71]. Furthermore, it is challenging yet important to improve the training efficiency while maintaining the substantial improvements.

CHAPTER 4: EXPLOITING MONOLINGUAL DATA AT SCALE

While target-side monolingual data has been proven to be very useful to improve neural machine translation (NMT) through back translation, source-side monolingual data is not well investigated. In this chapter, we study how to use both the source-side and target-side monolingual data for NMT, and propose an effective strategy leveraging both of them. First, we generate synthetic bitext by translating monolingual data from the two domains into the other domain using the models pretrained on genuine bitext. Next, a model is trained on a noised version of the concatenated synthetic bitext where each source sequence is randomly corrupted. Finally, the model is fine-tuned on the genuine bitext and a clean version of a subset of the synthetic bitext without adding any noise. Our approach achieves state-of-the-art results on WMT16, WMT17, WMT18 English↔German translations and WMT19 German→French translations, which demonstrate the effectiveness of our method. We also conduct a comprehensive study on how each part in the pipeline works.

4.1 INTRODUCTION

Neural machine translation (NMT) [5, 6, 7, 72] is well-known for its outstanding performance [30], which usually relies on large-scale bitext for training. However, high quality bitext is always limited and costly to collect. In contrast, there exists large amount of monolingual data which can be leveraged to augment the training corpus. How to effectively leverage monolingual data is an important research topic for NMT and there are plenty of works studying this problem [3, 24, 25, 35, 71, 73, 74, 75, 76, 77].

Among them, one of the most cited approach is back translation (BT) [3], which leverages the target-side monolingual data. Specifically, a target-to-source translation model (trained on the genuine bitext) is used to translate target-side monolingual sentences into the source domain to generate a set of synthetic bitext, which is then used together with the genuine bitext to train a source-to-target NMT model. While target-side monolingual data is well utilized by NMT through BT and its variants [3, 24, 29, 30], the investigation on source-side monolingual data is very limited. Only few attempts [25, 78] are made to explore source-side monolingual data, with a common high-level idea that a source-to-target translation model is trained to translate the source-side monolingual data into target domain, the resulted synthetic data is then used for further training. In this chapter, we study how to leverage both source-side and target-side monolingual data to boost the accuracy of NMT.

We propose a simple yet effective strategy to leverage two-side monolingual data for NMT,

which consists of three steps:

1. **Preparation**: We pretrain a source-to-target and a target-to-source NMT models on the genuine bitext, and use them to generate synthetic bitext by translating the monolingual data from the source/target domain to the other domain respectively.

2. **Large-scale Noised training**: The source sentences of the synthetic parallel corpus (including both the genuine source-side monolingual sentences and the synthetic source sentences back-translated from the target domain) are first corrupted. We then train an NMT model on this noised dataset together with the genuine bitext. We find that this step benefits from a large amount of monolingual data. The NMT model obtained from noised training can be further improved in the next finetune step.

3. **Clean training**: We randomly generate a subset of the clean synthetic bitext without adding any noise, and leverage them together with the genuine bitext to finetune the output model of step (2). This step only needs a small set of synthetic bitext.

We conduct a comprehensive study of our proposed method on WMT English↔German translation and German↔French translation and have the following empirical observations:

1. Using both source-side and target-side monolingual data is better than using monolingual data from only one domain (see Section 4.6.1).

2. Adding noise to large-scale synthetic bitext improves the accuracy of NMT (see Section 4.6.2 and Section 4.6.3).

3. Clean training/tuning of the model obtained from noised training further improves its accuracy (see Section 4.6.4).

4. Our method achieves state-of-the-art results on English↔German newstest 2016, 2017 and 2018 and German→French newstest 2019 (see Section 4.5).

4.2 BACKGROUND

In this chapter, our work is related to several important research directions of NMT, and we describe the previous relevant works in this section.

4.2.1 Neural Machine Translation

NMT adopts the sequence-to-sequence framework, which consists of an encoder and a decoder in the network architecture. The encoder and decoder are usually built upon deep neural networks, which can be recurrent neural network [4], convolutional neural network [6] or simple self-attention based transformer network [7]. The encoder encodes the source sentence into a continuous representation space, and the decoder will decodes the target sentence based on the encoder representations word-by-word. The objective of the NMT model training is to maximize the conditional probability of the target sentence given the source sequence. Different model architectures and modifications have been proposed to improve the training efficiency and NMT accuracy [30, 79, 80].

Different from the view of model architecture, in this chapter, we study the NMT training from the data perspective. Specifically, we study the effect in utilizing monolingual data in both source-side and target-side languages at scale, and investigate how to make the best utilization of the monolingual data.

4.2.2 Improving NMT by Monolingual Data

NMT heavily relies on large-scale parallel dataset for training. To augment the limited bilingual data, there are plenty of works attempt to leverage the monolingual data to help the training, which includes the language model fusion [73], back translation [3], dual learning [24, 71, 81] and self learning [25].

Gulcehre et al. [35] integrates the hidden states from the target language model into the NMT decoder to improve the accuracy. Sennrich et al. [3] propose the back translation (BT) approach to leverage the target-side monolingual data, which is simple and effective. BT requires training an additional target-to-source NMT model given the bilingual dataset, the model will be used to back translate the target-side monolingual data. The translation output and the target-side monolingual data then paired as synthetic parallel corpus to augment the original bilingual dataset in order to further train the source-to-target NMT model. Dual learning [24] extends the BT approach to train NMT systems in both translation directions. When jointly training the source-to-target and target-to-source NMT models, the two models can provide back translated data for each other direction and perform multi-rounds BT. This strategy is also successfully adopted to build the unsupervised translation system [31].

There are few attempts working on using the source-side monolingual data. Zhang and Zong [25] proposes self learning approach to generate the synthetic data for the source-side

monolingual data, which is a semi-supervised method. Wu et al. [34] leverages the source-side monolingual data to train the NMT system by learning reward function in a reinforcement learning framework.

4.2.3 Study of the Back Translation

Since BT is widely acknowledged and effective to improve the NMT model, there has been several works investigating back translation from different views. Poncelas et al. [26] studies on how using the back translated data as a training corpus (separately usage or combined with bilingual data) affects the performance of an NMT model. Burlot and Yvon [27] analyzes the accuracy impact from the quality of the BT data, the alternative uses of BT data to give explanations of why BT works. Cotterell and Kreutzer [28] provides an interpretation of back translation as approximate inference in a generative model of bitext and give a new algorithm. However, above studies are all based on the small-scale monolingual data, therefore it remains unclear in the large-scale setting.

Edunov et al. [29] firstly provides an extensive analysis of the back translation at scale. They investigate the different synthetic data generation methods, and also compare the different combination of the synthetic data with bitext. They finally build a strong system with millions of target-side monolingual sentences.

4.2.4 Data Augmentation in NMT

This chapter is also related to the general data augmentation methods in the NMT training. There are several common practices:

- Randomly swap words in nearby positions [82];
- Randomly drop words in a sentence [31];
- Randomly replace words with a placeholder token or with a sample from the uni-gram frequency distribution over the vocabulary [83]
- Randomly replace words [84] or word embedding [85] sampled from the output distribution of one language model.

Above noised data can simply augment the limited bilingual data, and have been proven to be effective.

4.3 APPROACH

In this section, we give a detailed introduction of our training strategy. Let X and Y denote two languages, and let $\mathcal{X}$ and $\mathcal{Y}$ denote two corresponding language sentence domains, which are the collection of all sentences. Let $\mathcal{B} = \{(x_i, y_i)\}_{i=1}^{N}$ denote the bilingual training pairs, where $x_i \in \mathcal{X}$, $y_i \in \mathcal{Y}$, N is the number of sentence pairs. Let $\mathcal{M}_x = \{x_j\}_{j=1}^{M_x}$ and $\mathcal{M}_y = \{y_j\}_{j=1}^{M_y}$ denote the collections of monolingual sentences, where M_x and M_y are the sizes of the two sets, $x_j \in \mathcal{X}$, $y_j \in \mathcal{Y}$. Our objective is to obtain a translation model $f : \mathcal{X} \mapsto \mathcal{Y}$, that can translate sequences from language X to language Y.

There are three steps of our training strategy: preparation, large-scale noised training and clean data tuning.

Step-1: Preparation. We first train two translation models $f_b : \mathcal{X} \mapsto \mathcal{Y}$ and $g_b : \mathcal{Y} \mapsto \mathcal{X}$ on the given bilingual data $\mathcal{B}$ by minimizing the negative log-likelihood:

$$\begin{aligned} &\min \sum_{(x,y)\in\mathcal{B}} -\log P(y|x; f_b); \\ &\min \sum_{(x,y)\in\mathcal{B}} -\log P(x|y; g_b). \end{aligned} \tag{4.1}$$

After that, we build the following two synthetic datasets through the trained models:

$$\begin{aligned} \bar{\mathcal{B}}_s &= \{(x, f_b(x))|x \in \mathcal{M}_x\}, \\ \bar{\mathcal{B}}_t &= \{(g_b(y), y)|y \in \mathcal{M}_y\}, \end{aligned} \tag{4.2}$$

where $\bar{\mathcal{B}}_s$ and $\bar{\mathcal{B}}_t$ can be seen the forward-translation (i.e., self learning) of source-side monolingual data [25] and back-translation of target-side monolingual data [3]. In practice, f_b and g_b will output one translation by either beam search or random sampling according to the model output distribution. Random sampling explores more possibility for the unknown generated sentence, which may benefit the data augmentation. In this work, we mainly adopt beam search to generate the sentences, but we will also compare different sequence generation methods in Section 4.6.2.

Step-2: Large-scale noised training. Inspired from the noised back translation [29] and denoising auto-encoder [44], we add noise to the source-side data of both $\bar{\mathcal{B}}_s$ and $\bar{\mathcal{B}}_t$ for training instead of directly using them to train models. We build two following noised

versions of the two augmented datasets,

$$\begin{aligned}\bar{\mathcal{B}}_s^n &= \{(\sigma(x), y) | (x, y) \in \bar{\mathcal{B}}_s\}, \\ \bar{\mathcal{B}}_t^n &= \{(\sigma(x), y) | (x, y) \in \bar{\mathcal{B}}_t\},\end{aligned} \tag{4.3}$$

where σ is the operator of adding noise. We follow [29] to design σ. Specifically,

- We randomly replace a word in the sentence to be a special token "<UNK>" (representing unknown words) with probability 0.1;
- We randomly drop the words in each position with probability 0.1;
- We randomly shuffle (swap) the words in the sentence, with constraint that the words will not be shuffled further than three positions distance.

We then train an NMT model f_n for X to Y translation on $\mathcal{B} \cup \bar{\mathcal{B}}_s^n \cup \bar{\mathcal{B}}_t^n$ by minimizing the negative log-likelihood:

$$\min \sum_{(x,y) \in \mathcal{B} \cup \bar{\mathcal{B}}_s^n \cup \bar{\mathcal{B}}_t^n} -\log P(y|x; f_n) \tag{4.4}$$

Compared with Edunov et al. [29], we enlarge the noised data from $\bar{\mathcal{B}}_t^n$ to $\bar{\mathcal{B}}_s^n \cup \bar{\mathcal{B}}_t^n$, instead of using the target-side monolingual data only. The intuition behind noised training is to force the encoder to discover more robust features and thus improve the generalization ability [44]. Besides, the output model of noised training has great potential to be improved in further finetune. Adding noise is widely acknowledged in other NLP tasks, like unsupervised NMT [31], BERT pretaining [40], etc. There are some other ways to augment/add noise to the training data [82] and we leave the combination with those approaches as future work. Compared to [29], we enlarge the noised data from $\bar{\mathcal{B}}_t$ to $\bar{\mathcal{B}}_s \cup \bar{\mathcal{B}}_t$, can further improve the generalization ability.

Step-3: Clean data tuning. After obtaining noised training model from step-2, we further finetune it on the clean version of the synthetic data without adding noise manually. For the efficiency, we can randomly subsample $\bar{\mathcal{B}}_s$ and $\bar{\mathcal{B}}_t$ to form the clean $\bar{\mathcal{B}}_s^s$ and $\bar{\mathcal{B}}_t^s$ dataset. Then we continue tuning f_n on the new datasets:

$$\min \sum_{(x,y) \in \mathcal{B} \cup \bar{\mathcal{B}}_s^s \cup \bar{\mathcal{B}}_t^s} -\log P(y|x; f) \tag{4.5}$$

where f is initialized by f_n.

There are different ways to obtain $\bar{\mathcal{B}}_s^s$ and $\bar{\mathcal{B}}_t^s$, such as the subset of $\bar{\mathcal{B}}_s$ and $\bar{\mathcal{B}}_t$ we described here. A more effective way is to train another $\bar{f}_b$ for X to Y translation and another $\bar{g}_b$ for Y to X translation, and use them to build new synthetic data for step-3. In this way, more diverse samples are included and we are able to achieve better results.

4.4 EXPERIMENTAL SETUP

We verify the effectiveness of our proposed training strategy in this section. We conduct experiments on four different translation tasks. We also make a comprehensive study about the effect of the monolingual data usage in our approach from various aspects.

4.4.1 Datasets

We carry out experiments on four different translation tasks from WMT19 competition[12], including En→De, De→En, De→Fr and Fr→De, where En, De and Fr are short for English, German and French respectively.

For En↔De translation tasks, the bilingual data consists of two parts:

- We concatenate "Europarl v9", "News Commentary v14", "Common Crawl corpus" and "Document-split Rapid corpus", remove the empty and duplicate lines and eventually get a clean dataset of news domain.

- We also use the Paracrawl dataset to extend the bilingual corpus. Considering Paracrawl is noisy, we apply a series of filtration rules to this dataset and remove the low-quality sentences, including sentences with too many punctuation marks or invalid characters, and those with too many or too few words, etc.

These two parts of data are then merged together to get the bilingual dataset. We eventually get a clean corpus with about $5M$ clean data and $18M$ Paracrawl data, which are denoted as `WMT` and `WMTPC` respectively for ease of reference.

The monolingual data we use is from newscrawl released by WMT19. We combine the newscrawl data from year 2016 to 2018 for the English and German monolingual corpus. After filtering with similar rules applied in Paracrawl (`preprocess_mono.py` in the supplementary document), an additional step is that we further perform language detection [86] on each side monolingual data. Finally, we randomly select about 120M sentences for each

[12] `http://www.statmt.org/wmt19/translation-task.html`

of English and German language. We choose newstest2015 as the validation set and use newstest2016 to newstest2019 as test sets.

For De↔Fr translation data, we follow the same process as that used in En↔De. Eventually, the `WMT` and ParaCrawl data of De↔Fr contains about 2M and 4.8M sentence pairs respectively. As for the French monolingual data, we use all the available newscrawl from previous years due to its small size compared to En and De. After filtering, we keep 60M monolingual data. We use the validation and test set provided by WMT19 for De↔Fr translation.

All datasets are tokenized with Moses [87] toolkit[13]. The vocabulary is built based on the Byte-Pair-Encoding (BPE) [88] with 35K merge operations[14] for both En↔De and De↔Fr datasets. We learn the BPE vocabulary jointly on the source and target language sentences.

4.4.2 Models

We choose the state-of-the-art Transformer network as our model structure, which consists of an encoder with 6 layers and a decoder with 6 layers. We use `transformer_big` configuration for all experiments: the dimension of word embedding and the inner feed-forward layer is 1024 and 4096 respectively. The parameters of source and target word embeddings, as well as the projection layer before softmax are shared. The number of attention heads is 16. The dropout is fixed as 0.2 due to validation performance. We conduct experiments on the fairseq [89] codebase[15].

4.4.3 Training

The models are trained by the Adam optimizer [90] with $\beta_1 = 0.9$, $\beta_2 = 0.98$. We use the default learning rate schedule used in [7] with the initial learning rate 5×10^{-4}. Label smoothing [91] is adopted with value 0.1. The batch size is set to be 4096 tokens per GPU. We use 4 P40 GPUs for training and update the parameters every 16 minibatches. The pretraining and the noised training both take about one week, and the finetune process takes about one day training.

To effectively leverage monolingual data, as mentioned in Section 4.3, we use two different groups of models to generate synthetic data for noised training and finetuning. For noised training, the models used for translating monolingual data are trained on `WMT`; for finetuning,

[13] `https://github.com/moses-smt/mosesdecoder/tree/master`
[14] `https://github.com/rsennrich/subword-nmt`
[15] `https://github.com/pytorch/fairseq`

Table 4.1: De-tokenized case-sensitive SacreBLEU on WMT En→De newstest2016, newstest2017, newstest2018, newstest2019 and the average score. "Avg" means the average BLEU score. "+" is conducted upon `WMTPC` dataset.

Model	**2016**	**2017**	**2018**	**2019**	**Avg**
`WMT`	34.0	28.0	41.3	37.3	35.15
`WMTPC`	37.1	30.5	45.6	40.3	38.38
+Noised Training	39.3	32.0	47.5	41.2	40.00
+Clean Tuning	**40.9**	**32.9**	**49.2**	**43.8**	**41.70**
`WMTPC`+BT	38.7	31.8	46.0	39.8	39.08

the models are trained on `WMTPC`. The intuition behind using different generation models is that each optimization stage can benefit from the new/unseen generated synthetic training data. Therefore, we adopt this way in our experiments. As for the data size, in noised training, $|\bar{\mathcal{B}}_s| = |\bar{\mathcal{B}}_t| = 60M$, and during finetuning, $|\bar{\mathcal{B}}_s^s| = |\bar{\mathcal{B}}_t^s| = 20M$.

4.4.4 Evaluation

To evaluate the model performance, we use beam search generation with beam width 5 and without length penalty. The BLEU score is measured by the de-tokenized case-sensitive SacreBLEU [92], which is widely adopted in the WMT competitions[16].

4.5 RESULTS

We summarize the results of our training strategy for En↔De in Table 4.1 and Table 4.2, and De↔Fr translations in Table 4.3 respectively.

As can be seen from Table 4.1 and Table 4.2, the Paracrawl dataset improves the model accuracy by a large margin. Compared with the baseline of using `WMT` only, on average, Paracrawl brings more than 3.0 and 4.0 BLEU improvements to En→De and De→En tasks respectively, indicating the effectiveness of leveraging more data. After large-scale noised training, the accuracy of each task is further boosted. On average, we improve the BLEU scores of En→De and De→En by 1.6 and 2.2 points, demonstrating the effectiveness of noised training. At last, the output models are tuned on the clean synthetic data. This step further brings 1.7 and 1.5 points gain over the previous step. For En↔De translation tasks,

[16] `https://github.com/mjpost/sacreBLEU`. SacreBLEU signatures: BLEU+case.mixed+lang.$task +numrefs.1+smooth.exp+test.$SET+tok.13a+version.1.2.12, where $task are en-de, de-en, de-fr and fr-de; $SET are 16, 17, 18, 19.

Table 4.2: De-tokenized case-sensitive SacreBLEU on WMT De→En newstest2016, newstest2017, newstest2018, newstest2019 and the average score. "Avg" means the average BLEU score. "+" is conducted upon WMTPC dataset.

Model	**2016**	**2017**	**2018**	**2019**	**Avg**
WMT	38.6	34.3	41.1	34.5	37.13
WMTPC	41.9	37.5	45.4	40.1	41.23
+Noised Training	46.1	39.8	47.7	40.2	43.45
+Clean Tuning	**47.5**	**41.0**	**49.5**	**41.9**	**44.98**
WMTPC+BT	45.8	39.8	47.2	38.6	42.90

Table 4.3: De-tokenized case-sensitive SacreBLEU on WMT De↔Fr newstest2019. "+" is conducted upon WMTPC dataset.

	De→Fr	**Fr→De**
WMT	31.2	26.1
WMTPC	34.2	28.6
+Noised Training	36.1	30.8
+Clean Tuning	**37.3**	**33.1**

we finally achieve 40.9, 32.9, 49.2 and 43.8 BLEU scores from newstest2016 to newstest2019; for De→En, the eventual BLEU scores of the four test sets end up at 47.5, 41.0, 49.5 and 41.9. Such strong improvements over the baseline reveal the great potential of our proposed strategy.

Prior to this work, the most common way to leverage monolingual data is BT [3]. We also compare our strategy to the vanilla BT, which consists of $20M$ synthetic data and WMTPC. The results are shown in the last row of Table 4.1 and Table 4.2. Without finetuning, our noised training strategy surpasses the BT by 0.92 and 0.55 points respectively. This shows that our strategy is more effective than standard BT.

In Table 4.3, we show the results of De↔Fr translation tasks. We have similar observation as that for En↔De translations. Specifically, the noised training can improve WMTPC by 1.9 and 2.2 points on De→Fr and Fr→De. When turning to clean tuning, we can obtain another 1.2 and 2.3 points improvement. The results on De↔Fr demonstrate that our strategy works across different languages.

Our method achieves state-of-the-art results on En↔De newstest2016, newstest2017 and newstest2018 and De→Fr newstest2019. We list several widely acknowledged systems on En→De translation in Table 4.4: MS-Marian [60], which is the champion of WMT18 En→De

Table 4.4: De-tokenized case-sensitive SacreBLEU on WMT En→De newstest2016, newstest2017 and newstest2018. MS-Marian and FAIR are ensemble results while ours are single-model results.

Model (En→De)	**2016**	**2017**	**2018**
FAIR (ensemble)	38.0	32.8	46.1
MS-Marian (ensemble)	39.6	31.9	48.3
Ours (single)	**40.9**	**32.9**	**49.2**

Table 4.5: De-tokenized case-sensitive SacreBLEU on WMT De→En newstest2016, newstest2017 and newstest2018. UCAM and RWTH are ensemble results while ours are single-model results.

Model (De→En)	**2016**	**2017**	**2018**
UCAM (ensemble)	45.1	38.7	48.0
RWTH (ensemble)	46.0	39.9	48.4
Ours (single)	**47.5**	**41.0**	**49.5**

competition, and FAIR's model [29] which leverages a large amount of monolingual data. Note that results of MS-Marian and FAIR are from ensemble models, while ours are from a single model. We find that our single model successfully surpasses the previous best systems in all test sets. Especially, for newstest2016 and newstest2018, we achieve 1.0 BLEU score improvement over the MS-Marian and set new records for these tasks. We also list the WMT18 top-2 systems for De→En translation in Table 4.5: RWTH [93] and UCAM [94] systems, which are both ensemble models. Similarly, our single model surpasses these ensemble systems by a large margin.

4.6 ANALYSIS

In this section, we provide a comprehensive study on the effect of monolingual data from different aspects, including data combination ways, data scale and the data generation methods. The experiments are conducted on the En→De translation and evaluated on newstest2016, newstest2017 and newstest2018. In this section, without specific clarification, the training data of each setting contains `WMTPC`.

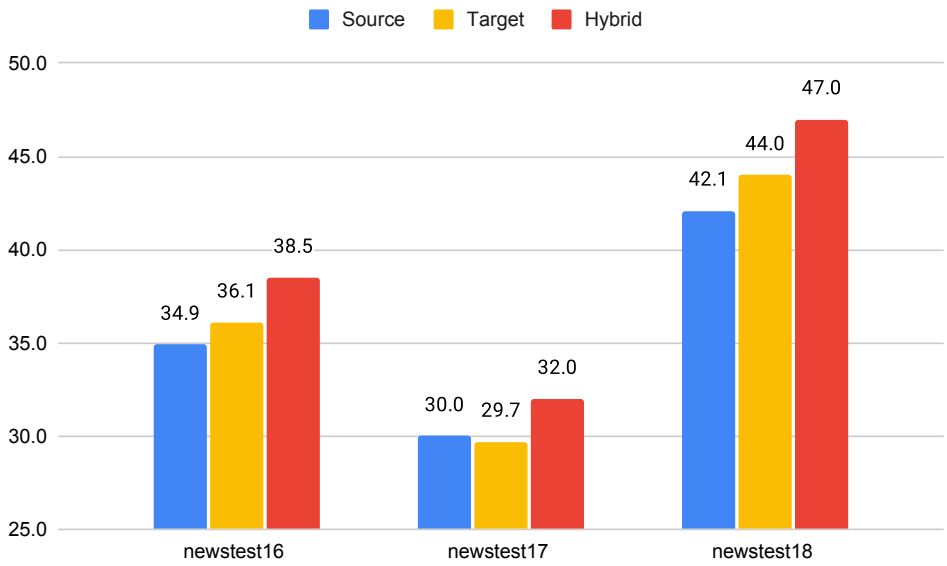


Figure 4.1: The de-tokenized SacreBLEU scores on En→De newstest2016, newstest2017 and newstest2018 of the models trained by different synthetic data: (1) Source: from source-side monolingual data $\bar{\mathcal{B}}_s$ only, (2) Target: from target-side monolingual data $\bar{\mathcal{B}}_t$ only and (3) Hybrid: the combination of source and target-side monolingual data $\bar{\mathcal{B}}_s + \bar{\mathcal{B}}_t$.

4.6.1 Source or Target Monolingual Data

We first investigate the effect about different combinations of monolingual data. Specifically, we compare three different ways to use monolingual data, including leveraging source-side monolingual data only (i.e., $\bar{\mathcal{B}}_s$), target-side monolingual data only (i.e., $\bar{\mathcal{B}}_t$), and the monolingual data from both two sides (i.e., $\bar{\mathcal{B}}_s + \bar{\mathcal{B}}_t$).

We keep the number of total synthetic data to be same across the three settings:

- *Source*: 120M source-side monolingual data $\bar{\mathcal{B}}_s$;
- *Target*: 120M target-side monolingual data $\bar{\mathcal{B}}_t$;
- *Hybrid*: the combination of 60M $\bar{\mathcal{B}}_s$ and 60M $\bar{\mathcal{B}}_t$.

We conduct this study on the clean synthetic data without noise, in order to verify which kind of data is more helpful for boosting performances.

The results are shown in Figure 4.1. From the figure, we can clearly observe that the best configuration is the hybrid one ($\bar{\mathcal{B}}_s + \bar{\mathcal{B}}_t$). With the same data size, on average, the model trained with hybrid synthetic dataset outperforms those trained with $\bar{\mathcal{B}}_s$ only and $\bar{\mathcal{B}}_t$ only by 2.6 and 3.5 points respectively. In particular, on newstest2018, the advantage of the hybrid dataset is nearly 5 points compared to $\bar{\mathcal{B}}_t$, which is extremely significant. This

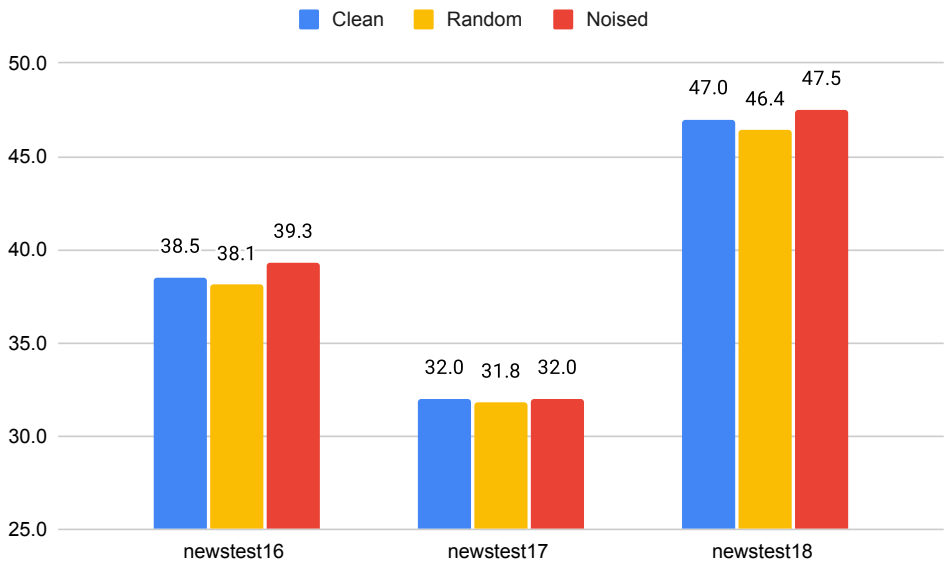


Figure 4.2: The de-tokenized SacreBLEU scores on En→De newstest2016, newstest2017 and newstest2018 of the models trained by synthetic data generated in different ways: (1) clean $\bar{\mathcal{B}}_s$ and $\bar{\mathcal{B}}_t$ data, (2) $\bar{\mathcal{B}}_s^r$ and randomly sampled $\bar{\mathcal{B}}_t^r$ data, and (3) noised $\bar{\mathcal{B}}_s^n$ and $\bar{\mathcal{B}}_t^n$ data.

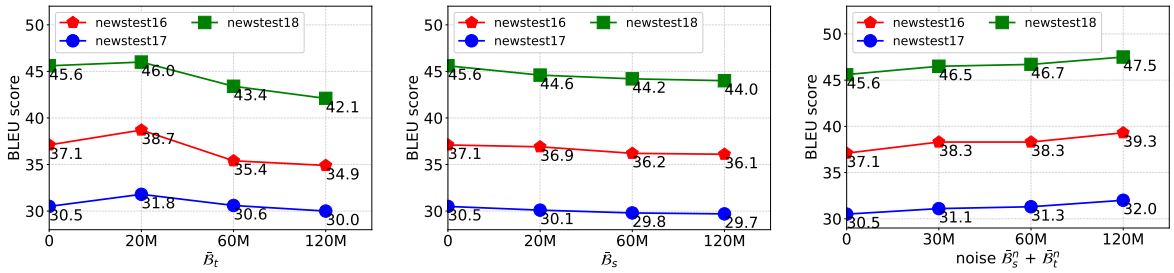
result strongly supports us that our training strategy by leveraging both the source-side and target-side monolingual data is helpful. Another point is that on such large-scale dataset, the $\bar{\mathcal{B}}_t$ data from back translation seems to be worse than the $\bar{\mathcal{B}}_s$ data generated by source-side monolingual data, which also supports the point that source-side monolingual is helpful.

4.6.2 Synthetic Data Generation

We conduct experiments on different generated synthetic data, to verify whether adding noise is essential. We use the same hybrid dataset (i.e., $60M$ $\bar{\mathcal{B}}_s$ and $60M$ $\bar{\mathcal{B}}_t$) as that used in Section 4.6.1 since the effectiveness has been verified.

We compare our noised training data $\bar{\mathcal{B}}_s^n$ and $\bar{\mathcal{B}}_t^n$ (*noised*) with two baselines, where the data size of each setting is kept 120M:

- *Clean*: $\bar{\mathcal{B}}_s$ and $\bar{\mathcal{B}}_t$ without any transformation;
- *Random*: $\bar{\mathcal{B}}_s$ and the randomly sampled synthetic data $\bar{\mathcal{B}}_t^r$, where each token in the translation is sampled from the multinomial distribution determined by the NMT model. Random sampling is an alternative way to introduce noise, where the resulted synthetic dataset is more diverse but the translation quality is relatively poor.



(a) Different scales of $\bar{\mathcal{B}}_t$ (b) Different scales of $\bar{\mathcal{B}}_s$ (c) Different scales of noised $\bar{\mathcal{B}}_s + \bar{\mathcal{B}}_t$

Figure 4.3: The de-tokenized SacreBLEU scores on newstest2016, newstest2017, newstest2018 of the models trained with varied data scales of (a) $\bar{\mathcal{B}}_t$ data, (b) $\bar{\mathcal{B}}_s$ data, and (c) hybrid $\bar{\mathcal{B}}_s + \bar{\mathcal{B}}_t$ data.

We present the result in Figure 4.2. Overall, noised traininig outperforms the clean training, e.g., 0.8 and 0.5 points advantage on newstest2016 and newstest2018 respectively. Compared with randomly sampled data, our noised data training achieves more than 1.0 BLEU improvement on newstest2016 and newstest2018, and similar performance on newstest2017. The above results suggest that our noised training is effective.

4.6.3 Scale of Monolingual Data

We present a comparison of different data scales for each type of synthetic data.

- *Scale of target-side data ($\bar{\mathcal{B}}_t$)*: We first look into the widely adopted back translation data $\bar{\mathcal{B}}_t$. We vary the data scale from 20M to 60M, which are obtained by random selection from the 120M corpus, and then to 120M.

  From Figure 4.3(a), we can observe that model performance is improved when we add 20M $\bar{\mathcal{B}}_t$ data. However, adding more clean back translated data starts to hurt the model performances. We suspect the reason is that the data distribution of $\bar{\mathcal{B}}_t$ shifts from the groundtruth distribution. Adding too much $\bar{\mathcal{B}}_t$ will enforce the model training to fit a biased distribution, making the generalization ability drop and thus, we observe that the BLEU scores drop. This result implies that we should choose a proper data size when using $\bar{\mathcal{B}}_T$ only, such as same size of bitext (e.g., 20M as we used).

- *Scale of source-side data ($\bar{\mathcal{B}}_s$)*: We then investigate the effect of data scale of source-side monolingual data, i.e., $\bar{\mathcal{B}}_s$ data. We also set the data sizes as {20M,60M,120M}.

  The results are shown in Figure 4.3(b). We do not observe any improvement over the bilingual system when adding $\bar{\mathcal{B}}_s$ data, and with more data added, the performance

slightly drops. It seems to be contrary to the experimental results in [25] which are conducted on RNN models. We speculate the self learning approach is actually hard for the model to boost itself, since there are no other signals to help the learning. The results demonstrate that leveraging source-side monolingual data alone is not a good choice. This phenomenon not only indicates the difficulty on how to leverage the source-side monolingual data, but also reveals that we find an effective way to combine both sides monolingual data to make them better utilized.

- *Scale of noised hybrid data ($\bar{\mathcal{B}}_s$ + $\bar{\mathcal{B}}_t$)* We finally study our noised training on different sizes of noised synthetic data. The experiments are conducted on the combination of noised 30M, 60M and 120M $\bar{\mathcal{B}}_s^n$ and $\bar{\mathcal{B}}_t^n$ with the same number of source-side and target-side monolingual sentences[17]. We choose the maximum data size as 120M due to GPU memory limitation. The results are shown in Figure 4.3(c).

We can see that the performances are consistently improved as the number of the noised synthetic data increases on all test sets. The result again proves that the hybrid usage of source-side and target-side monolingual data is an effective approach which outperforms using the two kinds of data individually.

In summary, first, we verify that source-side monolingual data is helpful and the best way to use it is to combine with target-side monolingual data. Next, we show that adding noise to synthetic data outperforms that without noise. Finally, we empirically prove that our strategy benefits from more monolingual data, while BT does not, demonstrating our strategy has great potential of utilizing more data.

4.6.4 Synthetic Tuning

We further study the clean tuning step in our training strategy. Two questions remain to be answered:

- Is it helpful to use two groups of models of building synthetic data for noised training and clean training as we discussed before?
- Is it helpful to use noised training first regarding future BLEU score achieved by the finetune step?

To answer these questions, we conduct another two experiments: (1) At finetuning step, use a subsample of synthetic data of noised training step, which is $\bar{\mathcal{B}}_s^s \subset \bar{\mathcal{B}}_s, \bar{\mathcal{B}}_t^s \subset \bar{\mathcal{B}}_t$. (2) Training without noise first and then finetuning using clean data.

[17]For example, same as hybrid 120M data, 60M hybrid data contains 30M $\bar{\mathcal{B}}_s^n$ and 30M $\bar{\mathcal{B}}_t^n$ data.

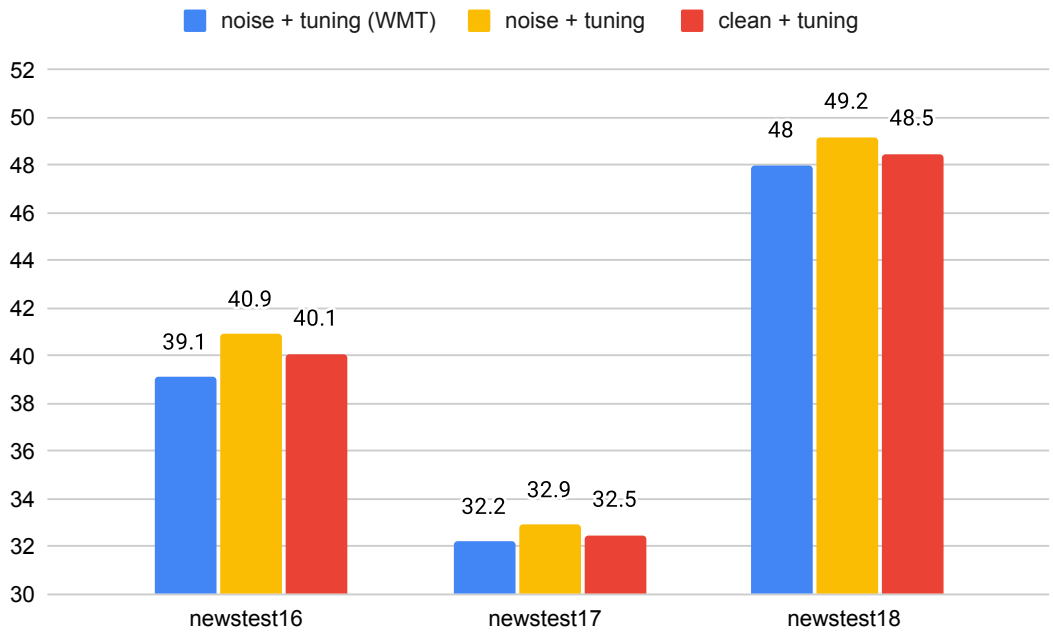


Figure 4.4: The de-tokenized SacreBLEU scores on En→De newstest2016, newstest2017 and newstest2018 of the models tuned by different synthetic data and pretrained models: (1) noised training model tuned on the subset of $\bar{\mathcal{B}}_s$ and $\bar{\mathcal{B}}_t$ (`WMT` in the figure), (2) noised training model tuned on the synthetic data as introduced in Section 4.4.3, and (3) clean training model tuned on the same synthetic data generated as (2).

We present the results in Figure 4.4. Obviously, we can see that tuning on the subset of $\bar{\mathcal{B}}_s$ and $\bar{\mathcal{B}}_t$ is worse than tuning on another set of synthetic data used in our experiments (e.g., 49.2 v.s. 48.0 on newstest2018). In addition, first training on the clean synthetic data and then tuning on other clean synthetic data also makes improvement, but similar to the results shown in Section 4.6.2, it is still not as good as our noised pretraining. Above results again prove the importance of our noised training and tuning on more diverse dataset is effective.

4.7 COMBINATION WITH MADL

In this section, we further study the performance of forward translation, back translation and MADL, and study the effective strategy of bring the best of the two strategies to further boost the performance of NMT systems.

4.7.1 Forward, Backward and Round-Trip Translation

In the Multi-Agent Dual Learning framework discussed in Chapter 3, we introduce a round-trip translation objective Equation (3.3), where the reconstruction error of the round-trip translation is used as feedback signals to regularize training. We show that with such round-trip translation objective, MADL can effectively utilize bitext and both source-side and target-side monolingual data.

We compare the performances of forward translation (FT), back translation (BT) and dual learning with round-trip translation (dual) in Section 3.4.3, under both bitext-only and the mix of bitext and monolingual data setting. In that small-scale study, we show that forward translation improves model performance more than back translation under the bitext-only setting, but does not work well with the monolingual data. Back translation can significantly boost the NMT performance with additional monolingual data. MADL outperforms both forward and back translation, and achieve thes state-of-the-art results.

In our large-scale noisy training pipeline, we utilize the monolingual data in both languages by combining the forward translation and back translation data as synthetic bitext. In the previous sections of this work, we show with extensive empirical study that forward and back translation, when combined together with the noisy training followed by clean fine-tuning strategy, can effectively utilize large-scale monolingual data to boost NMT performance.

In this section, we further explore the performance of forward translation, back translation and round-trip translation objective in MADL. We focus on the following two research questions:

- Which is the more effective strategy to leverage both source-side and target-side monolingual data at scale, round-trip translation objective or the combined forward and back translation?
- Can we bring the best of the two techniques together to further boost the performance of NMT systems?

4.7.2 Experiment Design

To answer the above research questions, we design additional experiments to compare the different techniques. Specifically, we compare the performances of the following systems:

- *Noisy training + Clean Fine-tuning*: We use the best combination from 4.6.4 as the baseline model of leveraging the combined forward and back translation data.

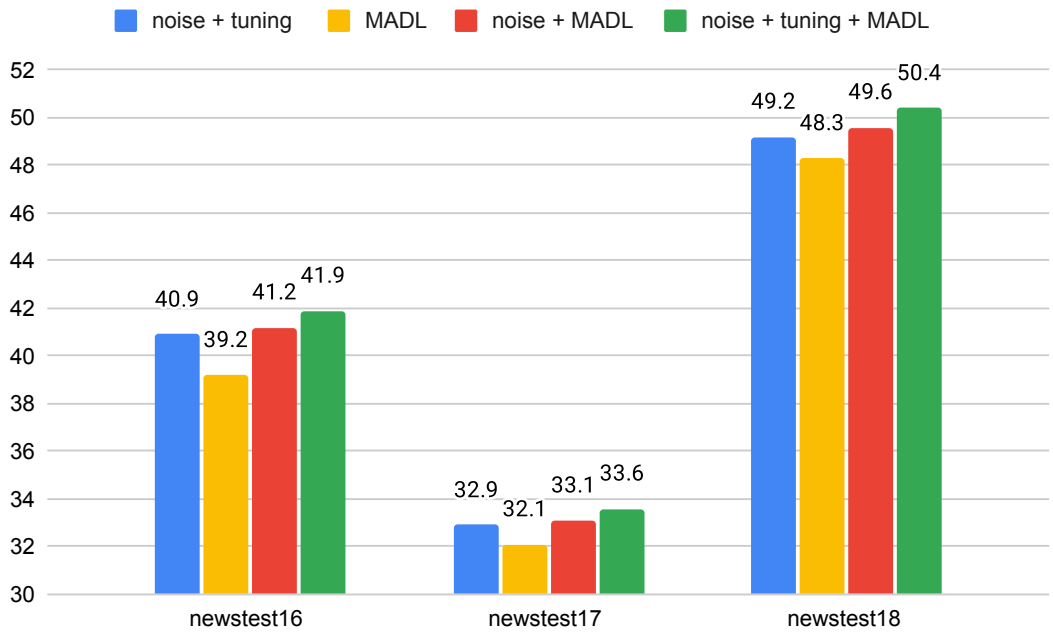


Figure 4.5: The de-tokenized SacreBLEU scores on En→De newstest2016, newstest2017 and newstest2018 of the models

- *MADL*: we train additional agents on the bitext data. And train the model with MADL following Algorithm 3.1 on the same monolingual data as noisy training.
- *Noisy training + MADL*: We first train the model and additional agent with the same noisy training step. Then, we continue training the model with MADL on the same data used in clean fine-tuning.
- *Noisy training + Clean Fine-tuning + MADL*: We first train the model and additional agent with the same noisy training followed by clean fine-tuning step. Then, we continue training the model with MADL on the same data used in clean fine-tuning.

4.7.3 Results

The results are illustrated in Table 4.5, from which we have the following observations:

1. Noisy training is the key step to large-scale training. We find that with the same amount of monolingual data, the noisy training followed by clean fine-tuning strategy proposed in this work outperforms MADL standalone. This is also in line with our previous observations in Section 4.6.4.

2. The model performs slightly better when fine-tuned with MADL than with standard fine-tuning.

3. The combination of forward, backward and round-trip translation can further boost the performance of NMT systems. We can see from the results that when combined together, the noisy training strategy and MADL can work together to bring further significant improvements to NMT models trained with large-scale data.

4.8 SUMMARY

In this work, we exploit the monolingual data at scale for the neural machine translation. Different from previous works which usually adopt back translation on target-side monolingual data only, we propose an effective training strategy to boost the NMT performance by leveraging both source-side and target-side monolingual data. Our approach contains three steps: synthetic data generation, large-scale noised training on synthetic data and clean data training/tuning. We verify our approach on the widely acknowledged WMT English↔German translation tasks and achieve state-of-the-art results, as well as that for the WMT German↔French translations. We also make a comprehensive study on the monolingual data utilization.

For future work, we would like to verify our training strategy on more language pairs and other sequence-to-sequence tasks. Furthermore, we are interested in studying our noised training with other data augmentation approaches.

CHAPTER 5: UTILIZING MONOLINGUAL DATA WITH MULTI-TASK LEARNING

While monolingual data has been shown to be useful in improving bilingual neural machine translation (NMT), effectively and efficiently leveraging monolingual data for Multilingual NMT (MNMT) systems is a less explored area. In this chapter, we introduce a multi-task learning (MTL) framework that jointly trains the model with the translation task on bitext data and two denoising tasks on the monolingual data. We conduct extensive empirical studies on MNMT systems with 10 language pairs from WMT datasets. We show that the proposed approach can effectively improve the translation quality for both high-resource and low-resource languages with large margin, achieving significantly better results than the individual bilingual models. We also demonstrate the efficacy of the proposed approach in the zero-shot setup for language pairs without bitext training data. Furthermore, we show the effectiveness of MTL over pre-training approaches for both NMT and cross-lingual transfer learning NLU tasks. MTL outperforms massive scale models trained on single task.

5.1 INTRODUCTION

Multilingual Neural Machine Translation (MNMT), which leverages a single NMT model to handle the translation of multiple languages, has drawn research attention in recent years [13, 14, 15, 16, 17]. MNMT is appealing since it greatly reduces the cost of training and serving separate models for different language pairs [13]. It has shown great potential in knowledge transfer among languages, improving the translation quality for low-resource and zero-shot language pairs [17, 18, 19].

Previous works on MNMT has mostly focused on model architecture design with different strategies of parameter sharing [15, 20, 21] or representation sharing [23]. Existing MNMT systems mainly rely on bitext training data, which is limited and costly to collect. Therefore, effective utilization of monolingual data for different languages is an important research question yet is less studied for MNMT.

Utilizing monolingual data (more generally, the unlabeled data) has been widely explored in various NMT and natural language processing (NLP) applications. Back translation (BT) [3], which leverages a target-to-source model to translate the target-side monolingual data into source language and generate pseudo bitext, has been one of the most effective approaches in NMT. However, well trained NMT models are required to generate back translations for each language pair, it is computationally expensive to scale in the multilingual setup. Moreover, it is less applicable to low-resource language pairs without adequate bitext

data. Self-supervised pre-training approaches [39, 40, 41, 42, 43], which train the model with denoising learning objectives on the large-scale monolingual data, have achieved remarkable performances in many NLP applications. However, catastrophic forgetting effect [47], where finetuning on a task leads to degradation on the main task, limits the success of continuing training NMT on models pre-trained with monolingual data. Furthermore, the separated pre-training and finetuning stages make the framework less flexible to introducing additional monolingual data or new languages into the MNMT system.

In this chapter, we propose a multi-task learning (MTL) framework to effectively utilize monolingual data for MNMT. Specifically, the model is jointly trained with translation task on multilingual parallel data and two auxiliary tasks: masked language modeling (MLM) and denoising auto-encoding (DAE) on the source-side and target-side monolingual data respectively. We further present two simple yet effective scheduling strategies for the multilingual and multi-task framework. In particular, we introduce a dynamic temperature-based sampling strategy for the multilingual data. To encourage the model to keep learning from the large-scale monolingual data, we adopt dynamic noising ratio for the denoising objectives to gradually increase the difficulty level of the tasks.

We evaluate the proposed approach on a large-scale multilingual setup with 10 language pairs from the WMT datasets. We study three English-centric multilingual systems, including many-to-English, English-to-many, and many-to-many. We show that the proposed MTL approach significantly boosts the translation quality for both high-resource and low-resource languages. Furthermore, we demonstrate that MTL can effectively improve the translation quality on zero-shot language pairs with no bitext training data. In particular, MTL achieves even better performance than the pivoting approach for multiple low-resource language pairs.

5.2 BACKGROUND

5.2.1 Neural Machine Translation

NMT adopts the sequence-to-sequence framework, which consists of an encoder and a decoder network built upon deep neural networks [4, 6, 7, 72]. The input source sentence is mapped into context representations in a continuous representation space by the encoder, which are then fed into the decoder to generate the output sentence. Given a language pair (x, y), the objective of the NMT model training is to maximize the conditional probability $P(y|x;\theta)$ of the target sentence given the source sentence.

NMT heavily relies on high-quality and large-scale bitext data. Various strategies have

been proposed to augment the limited bitext by leveraging the monolingual data. Back translation [3] utilizes the target-side monolingual data. Self learning [25] leverages the source-side monolingual data. Dual learning paradigms utilize monolingual data in both source and target language [24, 81, 95]. While these approaches can effectively improve the NMT performance, they have two limitations. First, they introduce additional cost in model training and translation generation, and therefore are less efficient when scaling to the multilingual setting. Second, back translation requires a good baseline model with adequate bitext data to start from, which limits its efficiency on low-resource settings.

5.2.2 Multilingual NMT

MNMT aims to train a single translation model that translates between multiple language pairs [13, 15]. Previous works explored the model architecture design with different parameter sharing strategies, such as partial sharing with shared encoder [14, 20], shared attention [15], task-specific attention [21], and full model sharing with language identifier [13, 16, 17]. There are also extensive studies on representation sharing that shares lexical, syntactic, or sentence level representations across different languages [18, 22, 23]. The models in these works rely on bitext for training, and the largely available monolingual data has not been effectively leveraged.

5.2.3 Self-supervised Learning

This chapter is motivated by the recent success of self-supervised learning for NLP applications [11, 39, 40, 41, 42, 43, 45]. Different denoising objectives have been designed to train the neural networks on large-scale unlabeled text. In contrast to previous work in pre-training with separated self-supervised pre-training and supervised finetuning stages, we focus on a multi-task setting to *jointly* train the MNMT model on both bitext and monolingual data.

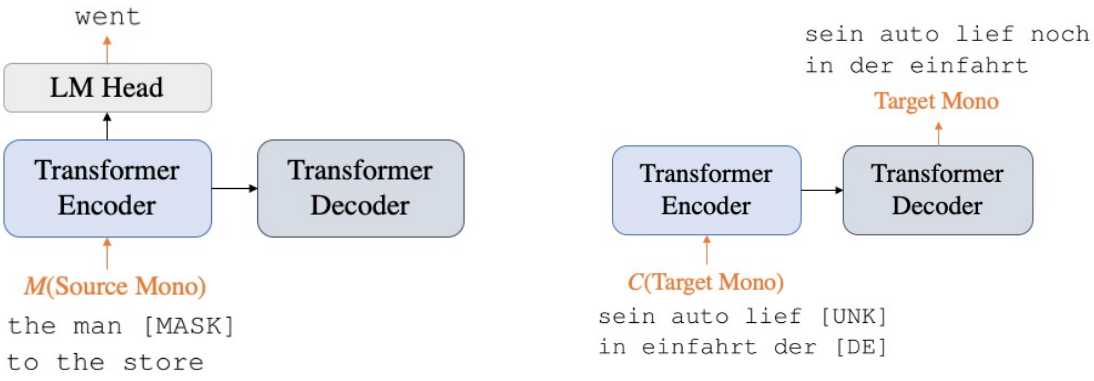
5.2.4 Multi-task Learning

Multi-task learning (MTL) [96], which trains the model on multiple related tasks to encourage representation sharing among different tasks and thus improve generalization performance, has been successfully used in many different machine learning applications [97, 98, 99]. In the context of NMT, MTL has been explored mainly to inject linguistic knowledge [100, 101, 102, 103, 104] with tasks such as part-of-speech tagging, dependency parsing,

semantic parsing, etc. In this chapter, we instead focus on auxiliary self-supervised learning tasks to leverage the monolingual data.

5.3 APPROACH

We propose a multi-task training framework to improve the MNMT system with monolingual data. We introduce the training tasks in Section 5.3.1 and scheduling strategies in Section 5.3.2.



(a) Masked Language Model (MLM)

(b) Denoising Auto-Encoding (DAE)

Figure 5.1: Illustration of the auxiliary tasks with monolingual data

5.3.1 Multi-task Learning

The main task in the MTL framework is the translation task trained on bitext corpora D_B of sentence pairs (x, y) with the cross-entropy loss:

$$\mathcal{L}_{MT} = \mathbb{E}_{(x,y)\sim D_B}[-\log P(y|x)] \tag{5.1}$$

With the large amount of monolingual data in different languages, we can train language models on both source-side [18] and target-side languages. We introduce two denoising language modeling tasks to help improve the quality of the translation model: the masked language model (MLM) task and the denoising auto-encoding (DAE) task.

[18]For the English-to-Many translation model, the source-side language is English; for Many-to-English and Many-to-Many, it refers the set of all other languages. Similarly for the target-side language.

Masked Language Model In the masked language model (MLM) task [40], sentences with tokens randomly masked are fed into the model and the model attempts to predict the masked tokens based on their context. MLM is beneficial for learning deep bidirectional representations. We introduce MLM as an auxiliary task to improve the quality of the encoder representations especially for the low-resource languages.

As is illustrated in Figure 5.1(a), we add an additional output layer to the encoder of the translation model and train the encoder with MLM on source-side monolingual data. The output layer is dropped during inference. The cross entropy loss for predicting the masked tokens is denoted as $\mathcal{L}_{MLM}$.

Following BERT [40], we randomly sample $R_M\%$ units in the input sentences and replace them with a special `[MASK]` token. A unit can either be a subword token, or a word consists of one or multiple subword tokens. We refer to them as token-level and word-level MLM.

Denoising Auto-Encoding (DAE) Denoising auto-encoding (DAE) [44] has been demonstrated to be an effective strategy for unsupervised NMT [11, 45]. Given a monolingual corpus D_M and a stochastic noising model C, DAE minimizes the reconstruction loss as shown in Eqn 5.2:

$$\mathcal{L}_{DAE} = \mathbb{E}_{x \sim D_M}[-\log P(x|C(x))] \tag{5.2}$$

As is illustrated in Figure 5.1(b), we train all model parameters with DAE on the target-side monolingual data. Specifically, we feed the target-side sentence to the noising model C and append the corresponding language ID symbol; the model then attempts to reconstruct the original sentence.

We introduce three types of noises for the noising model C.

- *Text Infilling* [42]: Following [43], we randomly sample $R_D\%$ text spans with span lengths drawn from a Poisson distribution ($\lambda = 3.5$). We replace all words in each span with a single blanking token.

- *Word Drop and Word Blank*: we randomly sample words from each input sentence, which are either removed or replaced with blanking tokens for each token position.

- *Word Swapping*: we slightly shuffle the order of words in the input sentence. Following [11], we apply a random permutation σ with condition $|\sigma(i) - i| \leq k, \forall i \in \{1, n\}$, where n is the length of the input sentence, and $k = 3$ is the maximum swapping distance.

Joint Training In the training process, the two self-learning objectives are combined with the cross-entropy loss for the translation task:

$$\mathcal{L} = \mathcal{L}_{MT} + \mathcal{L}_{MLM} + \mathcal{L}_{DAE} \tag{5.3}$$

In particular, we use bitext training data for the translation objective, source-side monolingual data for MLM, and target-side monolingual data for the DAE objective. A language ID symbol `[LID]` of the target language is appended to the input sentence in the translation and DAE tasks.

5.3.2 Task Scheduling

The scheduling of tasks and data associated with the task is important for multi-task learning. We further introduce two simple yet effective scheduling strategies in the MTL framework.

Dynamic Data Sampling One serious yet common problem for MNMT is data imbalance across different languages. Training the model with the true data distribution would starve the low-resource language pairs. Temperature-based batch balancing [17] is demonstrated to be an effective heuristic to ease the problem. For language pair l with bitext corpus D_l, we sample instances with probability proportional to $(\frac{|D_l|}{\sum_k |D_k|})^{\frac{1}{T}}$, where T is the sampling temperature.

While MNMT greatly improves translation quality for low-resource languages, performance deterioration is generally observed for high resource languages. One hypothesized reason is that the model might converge before well trained on high-resource data [105]. To alleviate this problem, we introduce a simple heuristic to feed more high-resource language pairs in the early stage of training and gradually shift more attention to the low-resource languages. To achieve this, we modify the sampling strategy by introducing dynamic sampling temperature $T(k)$ as a function of the number of training epochs k. We use a simple linear functional form for $T(k)$:

$$T(k) = \min\left(T_m, (k-1)\frac{T_m - T_0}{N} + T_0\right) \tag{5.4}$$

Where T_0 and T_m are the initial and maximum value for sampling temperature respectively. N is the number of warm-up epochs. The sampling temperature starts from a smaller value T_0, resulting in sampling leaning towards true data distribution. $T(k)$ gradually increases in

the training process to encourage over-sampling low-resource languages more to avoid them getting starved.

Dynamic Noising Ratio We further schedule the difficulty level of MLM and DAE from easier to more difficult. The main motivation is that training algorithms perform better when starting with easier tasks and gradually move to harder ones as promoted in curriculum learning [106]. Furthermore, increasing the learning difficulty can potentially help avoid saturation and encourage the model to keep learning from abundant data.

Given the monolingual data, the difficulty level of MLM and DAE tasks mainly depends on the noising ratio. Therefore, we introduce dynamic noising ratio $R(k)$ as a function of training steps:

$$R(k) = \min\left(R_m, (k-1)\frac{R_m - R_0}{M} + R_0\right) \tag{5.5}$$

Where R_0 and R_m are the lower and upper bound for noising ratio respectively and M is the number of warm-up epochs. Noising ratio R refers to the masking ratio R_M in MLM and the blanking ratio R_D of the *Text Infilling* task for DAE.

5.4 EXPERIMENTAL SETUP

5.4.1 Data

We evaluate MTL on a multilingual setting with 10 languages to and from English (En), including French (Fr), Czech (Cs), German (De), Finnish (Fi), Latvian (Lv), Estonian (Et), Romanian (Ro), Hindi (Hi), Turkish (Tr) and Gujarati (Gu).

Bitext Data The bitext training data comes from the WMT corpus. We concatenate all resources except WikiTitles provided by WMT of the latest available year and filter out duplicated pairs and pairs with the same source and target sentence. For Fr and Cs, we randomly sample 10M sentence pairs from the full corpus. The detailed statistics of bitext data can be found in Table 5.1.

We randomly sample $1,000$ sentence pairs from each individual validation set and concatenate them to construct a multilingual validation set. We tokenize all data with the SentencePiece model [107], forming a vocabulary shared by all the source and target languages with 32k tokens for all bilingual models (16k tokens for the low-resource languages Hi and Gu) and 64k tokens for the multilingual models.

Table 5.1: Statistics of the parallel resources from WMT. A list of 10 languages ranked with the size of bitext corpus translating to/from English.

Code	Language	#Bitext	Validation
Fr	French	10M	Newstest13
Cs	Czech	10M	Newstest16
De	German	4.6M	Newstest16
Fi	Finnish	4.8M	Newstest16
Lv	Latvian	1.4M	Newsdev17
Et	Estonian	0.7M	Newsdev18
Ro	Romanian	0.5M	Newsdev16
Hi	Hindi	0.26M	Newsdev14
Tr	Turkish	0.18M	Newstest16
Gu	Gujarati	0.08M	Newsdev19

Monolingual Data The monolingual data we use is mainly from NewsCrawl[19]. We apply a series of filtration rules to remove the low-quality sentences, including duplicated sentences, sentences with too many punctuation marks or invalid characters, sentences with too many or too few words, etc. We randomly select 5M filtered sentences for each language. For low-resource languages without enough sentences from NewsCrawl, we leverage data from CCNet [108].

Back Translation We use the target-to-source bilingual models to back translate the target-side monolingual sentences into the source domain for each language pair. The synthetic parallel data from back translation is mixed and shuffled with bitext and used together for the translation objective in training. We use the same monolingual data for back translation as the multi-task learning in all our experiments for fair comparison.

5.4.2 Model Configuration

We use Transformer for all our experiments using the PyTorch implementation[20] [89]. We adopt the `transformer_big` setting [7] with a 6-layer encoder and decoder. The dimensions of word embeddings, hidden states, and non-linear layer are set as 1024, 1024 and 4096 respectively, the number of heads for multi-head attention is set as 16. We use a smaller model setting for the bilingual models on low-resource languages Tr, Hi and Gu (with 3 encoder and decoder layers, 256 embedding and hidden dimension) to avoid overfitting and

[19] `http://data.statmt.org/news-crawl/`
[20] `https://github.com/pytorch/fairseq`

acquire better performance.

We study three multilingual translation scenarios including many-to-English (X→En), English-to-many (En→X) and many-to-many (X→X). For the multilingual model, we adopt the same Transformer architecture as the bilingual setting, with parameters fully shared across different language pairs. A target language ID token is appended to each input sentence.

For MLM, we randomly sample in total $R_M = 15\%$ units (tokens or words) from each input sentence, where 80% is replaced by the masking token, 10% is replaced by a random token and the rest 10% remains unchanged following Devlin et al. [40]. For DAE, we blank in total $R_D = 35\%$ words for text infilling, and set $\lambda = 3.5$ for Poisson distribution following Liu et al. [43]. The ratio for word blank and word drop is set as 20% and 10% respectively. We set $k = 3$ for word swapping.

5.4.3 Training and Evaluation

All models are optimized with Adam [90] with $\beta_1 = 0.9$, $\beta_2 = 0.98$. We set the learning rate schedule following Vaswani et al. [7] with initial learning rate 5×10^{-4}. Label smoothing [91] is adopted with 0.1. The models are trained on 8 V100 GPUs with a batch size of 4096 and the parameters are updated every 16 batches. During inference, we use beam search with a beam size of 5 and length penalty 1.0. The BLEU score is measured by the de-tokenized case-sensitive SacreBLEU[21] [92].

5.5 RESULTS

5.5.1 Main Results

We compare the performance of the bilingual models (*Bilingual*), multilingual models trained on bitext only, trained on both bitext and back translation (*+BT*) and trained with the proposed multi-task learning (*+MTL*). Translation results of the 10 languages translated to and from English are presented in Table 5.2 and 5.3 respectively. We can see that:

1. *Bilingual vs. Multilingual*: The multilingual baselines perform better on lower-resource languages, but perform worse than individual bilingual models on high-resource lan-

[21] SacreBLEU signatures: BLEU+case.mixed+lang.$l1-$l2 numrefs.1+smooth.exp+test.$SET+tok.13a+version.1.4.3, where $l1, $l2 are the language code (Table 5.1), $SET is the corresponding test set for the language pair.

Table 5.2: BLEU scores of 10 languages → English translation with bilingual, X→En and X→X systems. The languages are arranged from high-resource (left) to low-resource (right).

Test Set	Fr wmt15	Cs wmt18	De wmt18	Fi wmt18	Lv wmt17
Bilingual	36.2	28.5	40.2	19.2	17.5
X → En	34.6	28.0	39.7	20.1	19.6
+ MTL	36.4	31.5	42.3	23.0	22.1
+ BT	35.3	31.2	44.3	23.4	21.4
+ BT + MTL	35.3	31.9	45.4	23.8	22.4
X → X	33.9	28.1	39.0	19.9	19.5
+ MTL	35.1	29.6	40.1	21.7	21.3
+ BT	34.3	30.6	43.7	22.8	20.9
+ BT + MTL	35.3	31.2	43.7	23.1	21.5
Test Set	**Et wmt18**	**Ro wmt16**	**Hi wmt14**	**Tr wmt18**	**Gu wmt19**
Bilingual	19.7	29.8	14.1	15.1	9.3
X → En	23.9	33.2	20.5	21.3	16.1
+ MTL	28.7	37.9	24.8	25.7	22.3
+ BT	29.2	37.9	27.2	25.5	21.5
+ BT + MTL	30.5	39.1	28.7	27.6	23.5
X → X	24.5	33.7	22.4	22.0	17.2
+ MTL	27.3	36.8	23.9	25.2	23.3
+ BT	28.0	37.3	26.4	25.5	22.5
+ BT + MTL	29.5	38.1	27.5	26.2	23.4

guages like Fr, Cs and De. This is in concordance with the previous observations [17] and is consistent across the three multilingual systems (i.e., X→En, En→X and X→X).

2. *Multi-task learning*: Models trained with multi-task learning (+MTL) significantly outperform the multilingual baselines for all the languages pairs in all three multilingual systems, demonstrating the effectiveness of the proposed framework.

3. *Back Translation*: With the same monolingual corpus, MTL achieves better performance on some language pairs (e.g. Fr→En, Gu→En), while getting outperformed on some others, especially on the En→X direction. However, back translation is computationally expensive as it involves the additional procedure of training 10 bilingual models (20 for the X→X system) and generating translations for each monolingual sentence. Combining MTL with BT (+BT+MTL) introduces further improvements

Table 5.3: BLEU scores of English → 10 languages translation with bilingual, En→X and X→X systems. The languages are arranged from high-resource (left) to low-resource (right).

Test Set	Fr wmt15	Cs wmt18	De wmt18	Fi wmt18	Lv wmt17
Bilingual	36.3	22.3	40.2	15.2	16.5
En → X	33.5	20.8	39.0	14.9	18.0
+ MTL	33.8	21.7	39.8	15.2	18.5
+ BT	35.9	22.5	41.5	17.3	21.8
+ BT + MTL	36.1	23.6	42.0	17.7	22.4
X → X	32.2	19.4	37.3	14.5	17.5
+ MTL	33.3	20.9	39.2	15.6	19.3
+ BT	35.9	22.0	40.0	16.3	21.1
+ BT + MTL	35.8	22.4	41.2	16.9	21.7
Test Set	Et wmt18	Ro wmt16	Hi wmt14	Tr wmt18	Gu wmt19
Bilingual	15.0	23.0	12.2	13.3	7.9
En → X	19.8	25.5	12.4	15.7	11.9
+ MTL	21.1	26.5	16.1	17.6	15.4
+ BT	23.0	28.8	19.1	18.6	15.5
+ BT + MTL	24.0	29.8	19.8	19.4	17.8
X → X	19.6	25.4	13.9	16.3	12.0
+ MTL	21.1	26.8	16.5	18.1	15.5
+ BT	22.8	28.7	19.0	18.2	15.9
+ BT + MTL	23.2	29.7	19.2	18.7	16.0

for most language pairs without using any additional monolingual data. This suggests that when there is enough computation budget for BT, MTL can still be leveraged to provide good complementary improvement.

5.5.2 Zero-shot Translation

We further evaluate the proposed approach on zero-shot translation of non English-centric language pairs. We compare the performances of the pivoting method, the X→X baseline system, X→X with BT, and with MTL. For the pivoting method, the source language is translated into English first, and then translated into the target language [109, 110]. We evaluate on a group of high-resource languages with a multi-way parallel test set for De, Cs,

Table 5.4: Zero-shot translation performances on high-resource language pairs.

	De→Fr	Fr→De	Cs→De	De→Cs
Pivoting	22.1	19.1	17.5	15.9
X→X	15.1	11.9	15.5	15.2
+ BT	19.7	7.4	17.0	7.8
+ BT + MTL	20.1	12.2	19.7	12.0

Table 5.5: Zero-shot translation performances on low-resource language pairs.

	Et→Hi	Hi→Et	Hi→Tr	Tr→Hi
Pivoting	8.1	7.1	3.9	6.0
X→X	5.0	5.4	2.5	3.6
+ BT	4.9	5.5	4.5	5.7
+ BT + MTL	9.3	8.1	5.7	9.4
	Et→Tr	Tr→Et	Hi→Lv	Lv→Hi
Pivoting	7.1	7.8	8.6	7.9
X→X	7.8	8.8	6.1	4.7
+ BT	7.3	7.5	7.3	6.1
+ BT + MTL	7.8	10.5	8.2	8.6

Fr and En, constructed by newstest2009 with 3027 sentences and that of a group of low-resource languages Et, Hi, Tr and Hi (995 sentences). The results are shown in Table 5.4 and 5.5 respectively.

Utilizing monolingual data with MTL significantly improves the zero-shot translation quality of the X→X system, further demonstrating the effectiveness of the proposed approach. In particular, MTL achieves significantly better results than the pivoting approach on the high-resource pair Cs→De and almost all low-resource pairs. Furthermore, leveraging monolingual data through BT does not perform well for many low-resource language pairs, resulting in comparable and even downgraded performances. We conjecture that this is related to the quality of the back translations. MTL helps overcome such limitations with the auxiliary self-supervised learning tasks.

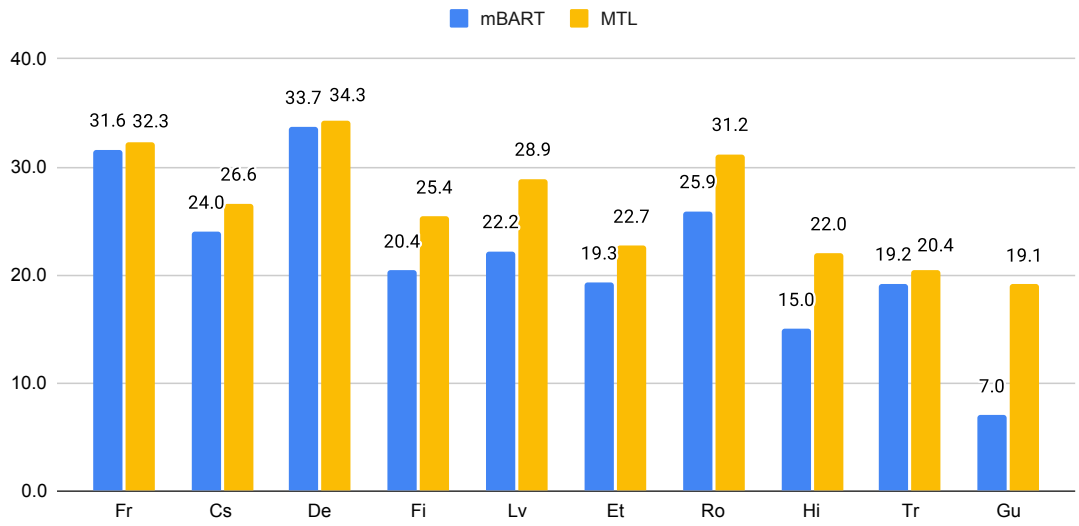


Figure 5.2: Comparison with mBART on En→X language pairs. BLEU scores are reported on the full individual validation set.

5.6 ANALYSIS

5.6.1 MTL vs. Pre-training

We also compare MTL with mBART [43], the state-of-the-art multilingual pre-training method for NMT. We adopt the officially released mBART model pre-trained on CC25 corpus[22] and finetune the model on the same bitext training data used in MTL for each language pair. As shown in Figure 5.2, MTL outperforms mBART on all language pairs. This suggests that in the scenario of NMT, jointly training the model with MT task and self-supervised learning tasks could be a better task design than the separated pre-training and finetuning stages. It is worth noting that mBart is utilizing much more monolingual data; for example, it uses 55B English tokens and 10B French tokens, while our approach is using just 100M tokens each. This indicates that MTL is more data efficient.

5.6.2 Multi-task Objectives

We present ablation study on the learning objectives of the multi-task learning framework. We compare performance of multilingual baseline model with translation objective only, jointly learning translation with MLM, jointly learning translation with DAE, and the combination of all objectives. Table 5.6 shows the results on a high-resource pair De↔En

[22] `https://dl.fbaipublicfiles.com/fairseq/models/mbart/mbart.CC25.tar.gz`

Table 5.6: Comparison of different multi-task learning objectives on De-En and Tr-En translation. BLEU scores are reported on the full individual validation set.

	De→En		**Tr→En**	
Models	X→En	X→X	X→En	X→X
Multilingual	36.5	36.1	20.0	20.9
+ MLM	36.3	36.6	21.0	21.4
+ DAE	37.7	37.8	21.7	22.6
+ MLM + DAE	38.7	37.6	22.9	23.7
	En→De		**En→Tr**	
Models	En→X	X→X	En→X	X→X
Multilingual	33.0	32.0	16.4	17.0
+ MLM	32.9	32.6	16.9	17.2
+ DAE	33.7	33.7	17.3	18.2
+ MLM + DAE	34.2	33.6	18.0	18.3

and low-resource pair Tr↔En. We can see that introducing MLM or DAE can both effectively improve the performance of multilingual systems, and the combination of both yields the best performance. We also observe that MLM is more beneficial for X→En compared with En→X direction, especially for the low-resource languages. This is in concordance with our intuition that the MLM objective contributes to improving the encoder quality and source-side language modeling for low-resource languages.

5.6.3 Dynamic Sampling Temperature

We study the effectiveness of the proposed dynamic sampling strategy. We compare multilingual systems using a fixed sampling temperature $T = 5$ with systems using dynamic temperature $T(k)$ defined in Equation 5.4. We set $T_0 = 1, T_m = 5, N = 5$, which corresponds to gradually increasing the temperature from 1 to 5 with 5 training epochs and saturate to $T = 5$ afterwards. The results for X→En and En→X systems are presented in Figure 5.3 and 5.4 respectively, where we report ΔBLEU relative to their corresponding bilingual baseline model that was evaluated on the individual validation sets for each language pairs. The dynamic temperature strategy improves the quality for high-resource language pairs (e.g. Fr→En, De→En, En→Fr), while introducing minimum effect for mid-resource languages (Lv). Surprisingly, the proposed strategy also greatly boosts performance for low-resource languages Tr and Gu, with over +1 BLEU gain for both to and from English direction.

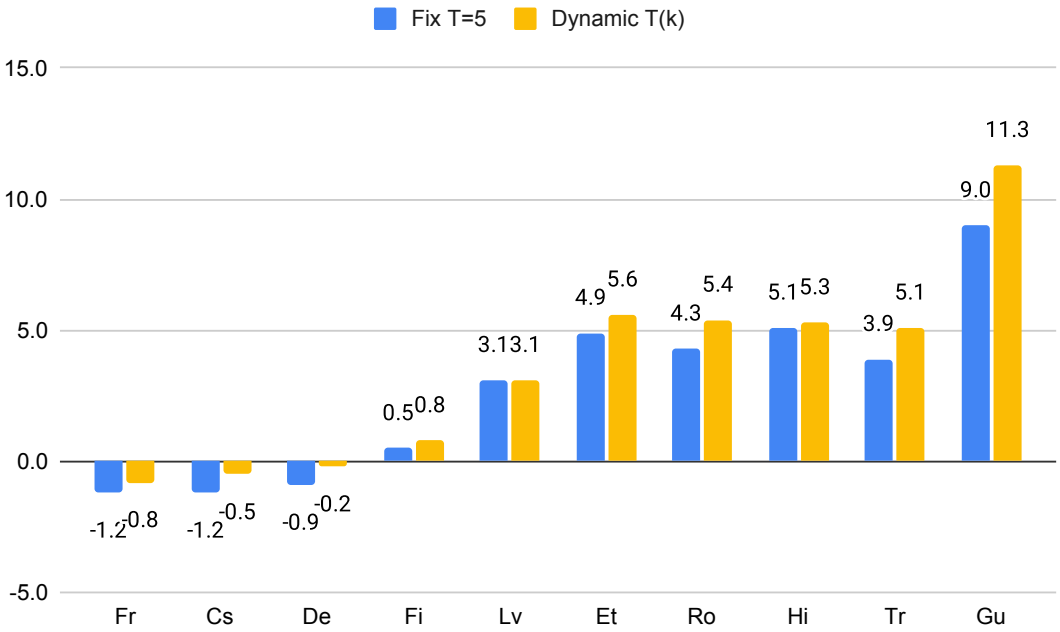


Figure 5.3: Performance gain of data sampling strategies on the X→En system. Results are reported as ΔBLEU relative to the corresponding bilingual baseline on validation sets. The languages are arranged from high-resource (left) to low-resource (right).

5.6.4 Noising Scheme

We study the effect of different noising schemes in the MLM and DAE objectives. As introduced in Section 5.3.1, we have token-level and word-level masking scheme for MLM depending on the unit of masking. We also have two noising schemes for DAE, where the *Text Infilling* task blanks a span of words (span-level), and the *Word Blank* task blanks the input sentences at word-level. We compare performance of these different noising schemes on X→En system as shown in Figure 5.5.

We report ΔBLEU relative to the multilingual X→En baseline on the corresponding language pairs for each noising scheme. As we can see, the model benefits most from the word-level MLM and the span-level *Text Infilling* task for DAE. This is in concordance with the intuition that the *Text Infilling* task teaches the model to predict the length of masked span and the exact tokens at the same time, making it a harder task to learn. We use the word-level MLM and span-level DAE as the best recipe for our MTL framework.

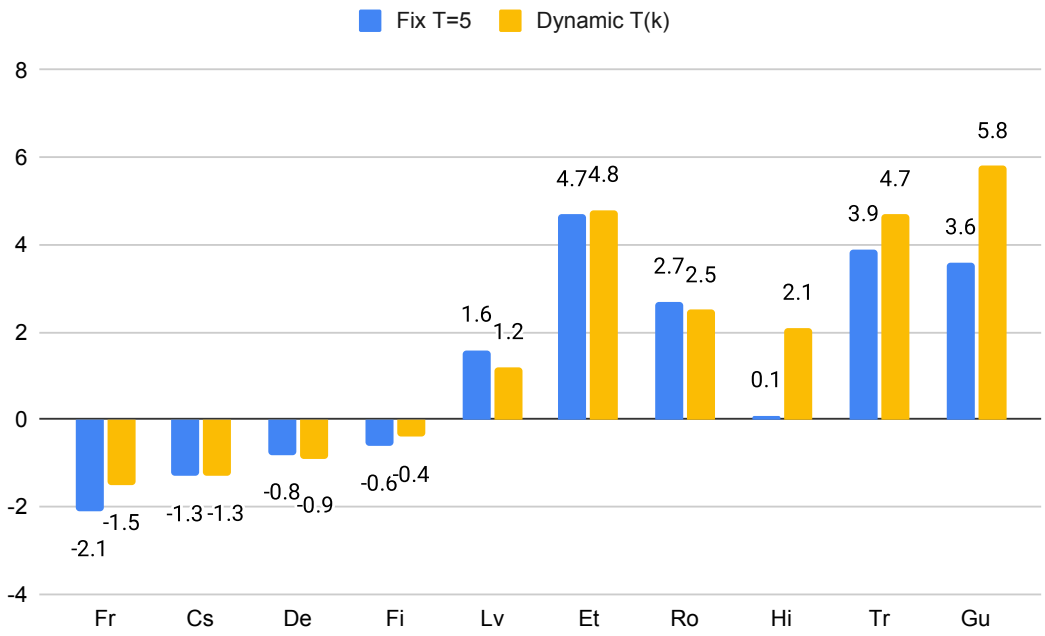


Figure 5.4: Performance of data sampling strategies on the En→X system. Results are reported as ΔBLEU relative to the corresponding bilingual baseline on validation sets. The languages are arranged from high-resource (left) to low-resource (right).

5.6.5 Noising Ratio Scheduling

In our initial experiments, we found that the dynamic noising ratio strategy does not effectively improve the performance. We suspect that it is due to the limitation of data scale. We experiment with a larger scale setting by increasing the amount of monolingual data from 5M sentences for each language to 20M. For low-resource languages without enough data, we take the full available amount (18M for Lv, 11M for Et, 5.2M for Gu).

Table 5.7 shows results on X→En MNMT model with large-scale monolingual data setting. We compare the performance of multilingual with back translation baseline, a model with MTL and a model with both MTL and dynamic noising ratio. For the dynamic noising ratio, we set the masking ratio for MLM to increase from 10% to 20% and blanking ratio for DAE to increase from 20% to 40%.

As we can see, the dynamic noising strategy helps boost performance for mid-resource languages like Lv and Et, while introducing no negative effect to other languages. For future study, we would like to cast the dynamic noising ratio over different subsets of monolingual datasets to prevent the model from learning to copy and memorize.

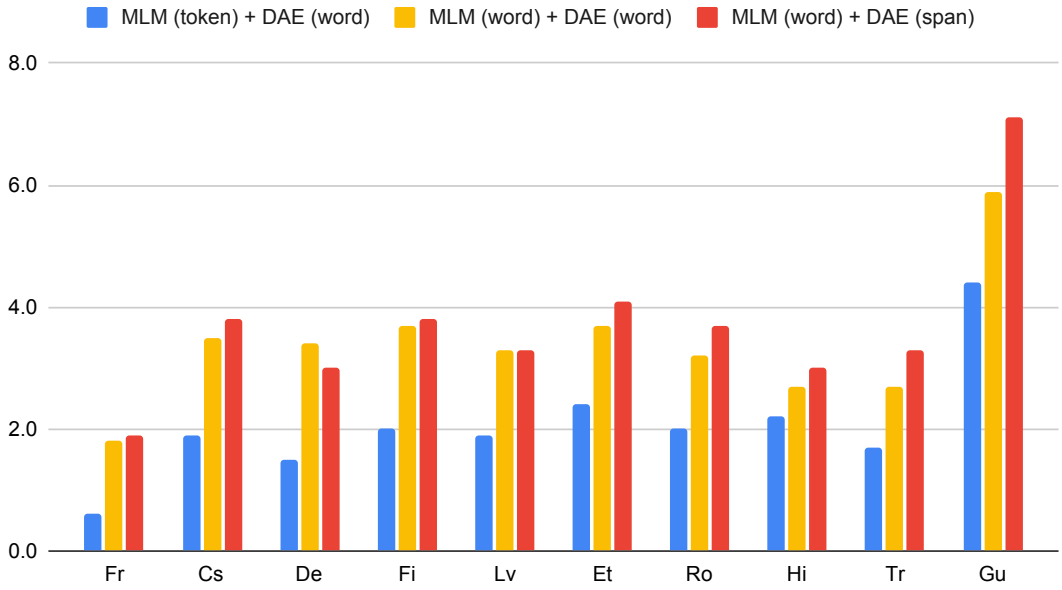


Figure 5.5: Performance of different noising schemes on the X→En system. Results are reported as ΔBLEU relative to the multilingual X→En baseline on validation sets. The languages are arranged from high-resource (left) to low-resource (right).

Table 5.7: BLEU scores of dynamic noising strategy on X→En translation system with large-scale monolingual data setting on validation sets.

	De	Lv	Et	Hi	Tr
Bilingual	32.9	23.4	18.7	12.9	16.1
X→En + BT	35.2	30.1	28.3	18.7	24.7
+ MTL	36.9	31.9	31.4	21.6	27.4
+ Dynamic	37.0	32.4	32.0	21.7	27.5

5.6.6 MTL for Cross-Lingual Transfer Learning for NLU

Large scale pre-trained cross-lingual language models such as mBERT [40] and XLM-Roberta [111] are the state-of-the-art for cross-lingual transfer learning on natural language understanding (NLU) tasks, such as XNLI [112] and XGLUE [113]. Such models are trained on massive amount of monolingual data from all language as a masked language model. It has been shown that massive MNMT models are not able to match the performance of pre-trained language models such as XLM-Roberta on NLU downstream tasks [114]. In [114], the MNMT models are massive scale models trained only on the NMT task. They are not able to outperform XLM-Roberta, which is trained with MLM task without any parallel data. In this work, we evaluate the effectiveness of our proposed MTL approach for cross-

Table 5.8: Evaluation on XNLI task, XLM-Roberta results is our reproduction of the results. Massively Multilingual Translation Encoder (MMTE) is reported from [114]

	EN	ES	DE	FR
XLM-Roberta	84.7	79.4	77.4	79.1
MMTE	79.6	71.6	68.2	69.5
MTL	84.8	80.1	78.3	79.8

Table 5.9: Evaluation on XGLUE NER task, XLM-Roberta results is our reproduction of the results.

	EN	ES	DE
XLM-Roberta	91.1	76.5	70.3
MTL	91.2	77.0	75.0

lingual transfer leaning on NLU application. Intuitively, MTL can bridge this gap since it utilizes NMT, MLM and DAE objectives.

In the experiment, we train a system on 6 languages using both bitext and monolingual data. For the bitext training data, we use 30M parallel sentences per language pair from in-house data crawled from the web. For the monolingual data, we use 40M sentences per language from CCNet [108]. Though this is a relatively large-scale setup, it only leverages a fraction of the data used to train XLM-Roberta for those languages. We train the model with 12 layers encoders and 6 layers decoder. The hidden dimension is 768 and the number of heads is 8. We tokenize all data with the SentencePiece model [107] with the vocabulary size of 64K. We train a many-to-many MNMT system with three tasks described in Section 5.3.1: NMT, MLM, and DAE. Once the model is trained, we use the encoder only and discard the decoder. We add a feedforward layer for the downstream tasks.

As shown in Table 5.8, MTL outperform both XLM-Roberta and MMTE [114] which are trained on massive amount of data in comparison to our system. XLM-Roberta is trained only on MLM task and MMTE is trained only on NMT task. Our MTL system is trained on three tasks. The results clearly highlight the effectiveness of multi-task learning, and demonstrate that it can outperform single-task systems trained on massive amount of data. We observe the same pattern in Table 5.9 with XGLUE NER task, which outperforms SOTA XLM-Roberta model.

5.7 SUMMARY

In this work, we propose a multi-task learning framework that jointly trains the model with the translation task on bitext data, the masked language modeling task on the source-side monolingual data and the denoising auto-encoding task on the target-side monolingual data. We explore data and noising scheduling approaches and demonstrate their efficacy for the proposed approach. We show that the proposed MTL approach can effectively improve the performance of MNMT on both high-resource and low-resource languages with large margin, and can also significantly improve the translation quality for zero-shot language pairs without bitext training data. We showed that the proposed approach is more effective than pre-training followed by finetuning for NMT. Furthermore, we showed the effectiveness of multitask learning for cross-lingual downstream tasks outperforming SOTA larger models trained on single task.

For future work, we are interested in investigating the proposed approach in a scaled setting with more languages and a larger amount of monolingual data. Scheduling the different tasks and different types of data would be an interesting problem. Furthermore, we would also like to explore the most sample efficient strategy to add a new language to a trained MNMT system.

CHAPTER 6: CONCLUSIONS AND OUTLOOK

In this thesis, we focus on exploiting large-scale monolingual data to improve the performance of NMT systems, and explore two research questions: (1) How to utilize both source-side and target-side monolingual data for NMT; and (2) How to leverage the available bitext and monolingual data in the most effective and data efficient way. To this end, we propose three novel and effective solutions.

- **Multi-Agent Dual Learning**: In Chapter 3, we exploit the potential of monolingual data with the dual learning paradigm, which introduces a round-trip translation objective to regularize the learning process. We propose Multi-Agent Dual Learning (MADL) to further improve data utilization and translation quality. In MADL, we introduce additional primal models that learn the source-to-target translation, and dual models that learn the target-to-source translation into the standard dual learning system. The additional models, with similar capability and certain level of diversity, work together to provide more reliable and robust feedback signals to enhance the learning process of the main models. MADL can effectively leverage bitext and monolingual data in both source and target languages.

- **Large-scale Noisy Training**: In Chapter 4, we study the effect of forward translation and back translation data, and propose a large-scale noisy training pipeline to effectively utilize both of them. First, we generate synthetic bitext with the mix of forward and back translation. Next, we train the NMT model on a noisy version of this synthetic corpus, where each source sentence is randomly corrupted. Finally, the model is fine-tuned on the genuine bitext and a clean, high-quality subset of the synthetic bitext without any noise. Our strategy overcomes the previous problem that the model performance drops when training with forward translation data or too much synthetic data, and demonstrate an effective solution on the hybrid use of both source and target-side monolingual data.

- **Multi-task Learning**: In Chapter 5, we further explore a more general scenario beyond bilingual NMT for rich-resource language pairs. We focus on Multilingual NMT, which handle the translation of multiple languages in both high-resource and low-resource with a single NMT model. We propose a multi-task learning framework, which jointly trains the model with the translation task on bitext data, and two denoising tasks on the source and target-side monolingual data respectively. Our approach is more computationally efficient than previous work utilizing round-trip, forward and

backward translations, since no additional NMT models or generation steps is needed, making it more applicable to multilingual NMT and low-resource language translations. Moreover, our approach bypasses the problem of catastrophic forgetting effect that limits the success of self-supervised learning for NMT. It is demonstrated to be a data efficient framework that effectively combine bitext and monolingual data for multilingual NMT.

Our three lines of work, MADL with a round-trip translation objective, large-scale noisy training combining forward and backward translation, and multi-task learning with self-supervised denoising objectives, has provided effective solutions to exploit monolingual data for NMT. The relation of these three line of work are as follows:

First, the first two lines of work, namely, MADL with round-trip translation objective introduced in Chapter 3, and the combined usage of forward and backward translation introduced in Chapter 4, can both effectively utilize source and target-side monolingual data to improve the accuracy of NMT models. In Section 4.7, we further discuss the comparison and potential combination of the two approaches. We find that in the more scaled setting with over 100M monolingual sentences in both languages, noisy training is the key factor to success. We show that the combination of the two approaches, i.e., large-scale noisy training with the mix of forward and back translation, followed by clean fine-tuning and MADL, achieve the best performance.

Furthermore, we further study a more general setting of multilingual NMT with both high-resource and low-resource languages. The first two lines of work is computationally expensive to scale in such multilingual setup, since multiple well trained NMT models are required to generate forward and back translations for each language pair, or serve as additional agents in MADL. Moreover, it is less applicable to low-resource language pairs without adequate bitext data. The multi-task learning approach with self-supervised denoising objectives introduced in Chapter 5, overcome the aforementioned limitations of the first two lines of work, and is demonstrated to be a general solution to effectively utilize monolingual data in the multilingual setup. The denoising objectives play a key role in the multi-task learning framework, which is in concordance with our observation on the importance of noisy training in Chapter 4.

While great breakthrough has been made in the field of NMT, and most NMT systems are providing high quality, even human-parity performance, in high resource language pair and major domains, there are still challenges remaining. Recent research breakthrough in Large Language Models (LLMs) such as GPT-3, GPT-4 and PaLM [115, 116, 117] is posting a significant impact in the field of natural language processing. LLMs pre-train the large

Transformer-based neural networks with self-supervised learning (discussed in Section 2.3.4) with tremendous resources. LLMs provide a solution to effectivly leverage unlabeled data at scale, and demonstrate emergent abilities in various complex understanding, reasoning and generation tasks. How to effectively leverage and adopt LLMs for NMT to further push the frontier of NMT open up future research opportunities in many different directions.

- *Multilingual NMT*: Multilingual NMT has always been challenging for low-resource language pairs, especially for the out-of-English (En→X) translation and zero-shot X→Y pairs of non-centric languages. What would be the most data efficient strategy to train the MNMT model with hundreds of languages, how to effectively leverage pre-trained NMT models or LLMs with few-shot translation samples would be an interesting direction to further explore.

- *Customized NMT*: Customized translation is an important application scenario. Certain attributes of the translations generated in the target language are controlled given the customized requirements, such as politeness, formality or domain of the vocabulary [118, 119, 120]. How to adopt the pre-trained NMT or large language models with constraints in a data efficient manner remains an interesting yet challenging research question.

- *Translate, Critic and Edit*: LLMs have demonstrated strong general purpose reasoning and knowledge-based capabilities, and could potentially be leveraged for quality evaluation, error analysis and post editing for NMT systems.

Looking at the big picture, we believe our exploration in leveraging monolingual data to improve NMT systems has provided effective solutions and insight to the field of NMT. With the emergent ability of the recent LLMs, more potentials and opportunities have been opened up, and more breakthroughs will be made in creating a more intelligent and reliable neural machine translation system.

REFERENCES

[1] Y. Xia, T. Qin, W. Chen, J. Bian, N. Yu, and T.-Y. Liu, "Dual supervised learning," in *International Conference on Machine Learning*, 2017.

[2] Y. Xia, X. Tan, F. Tian, F. Gao, D. He, W. Chen, Y. Fan, L. Gong, Y. Leng, R. Luo *et al.*, "Microsoft research asia's systems for wmt19," in *Proceedings of the Fourth Conference on Machine Translation*, 2019, pp. 424–433.

[3] R. Sennrich, B. Haddow, and A. Birch, "Improving neural machine translation models with monolingual data," in *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*, 2016, pp. 86–96.

[4] I. Sutskever, O. Vinyals, and Q. V. Le, "Sequence to sequence learning with neural networks," in *Advances in neural information processing systems*, 2014, pp. 3104–3112.

[5] Y. Wu, M. Schuster, Z. Chen, Q. V. Le, M. Norouzi, W. Macherey, M. Krikun, Y. Cao, Q. Gao, K. Macherey, J. Klingner, A. Shah, M. Johnson, X. Liu, Łukasz Kaiser, S. Gouws, Y. Kato, T. Kudo, H. Kazawa, K. Stevens, G. Kurian, N. Patil, W. Wang, C. Young, J. Smith, J. Riesa, A. Rudnick, O. Vinyals, G. Corrado, M. Hughes, and J. Dean, "Google's neural machine translation system: Bridging the gap between human and machine translation," *CoRR*, 2016. [Online]. Available: http://arxiv.org/abs/1609.08144

[6] J. Gehring, M. Auli, D. Grangier, D. Yarats, and Y. N. Dauphin, "Convolutional sequence to sequence learning," in *International Conference on Machine Learning*, 2017.

[7] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in neural information processing systems*, 2017.

[8] J. L. Ba, J. R. Kiros, and G. E. Hinton, "Layer normalization," *arXiv preprint arXiv:1607.06450*, 2016.

[9] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[10] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: a simple way to prevent neural networks from overfitting," *The Journal of Machine Learning Research*, vol. 15, no. 1, pp. 1929–1958, 2014.

[11] G. Lample, M. Ott, A. Conneau, L. Denoyer, and M. Ranzato, "Phrase-based & neural unsupervised machine translation," in *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, 2018, pp. 5039–5049.

[12] P. Bojanowski, E. Grave, A. Joulin, and T. Mikolov, "Enriching word vectors with subword information," *Transactions of the Association for Computational Linguistics*, 2017.

[13] M. Johnson, M. Schuster, Q. V. Le, M. Krikun, Y. Wu, Z. Chen, N. Thorat, F. Viégas, M. Wattenberg, G. Corrado *et al.*, "Google's multilingual neural machine translation system: Enabling zero-shot translation," *Transactions of the Association for Computational Linguistics*, pp. 339–351, 2017.

[14] D. Dong, H. Wu, W. He, D. Yu, and H. Wang, "Multi-task learning for multiple language translation," in *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics*, 2015, pp. 1723–1732.

[15] O. Firat, K. Cho, and Y. Bengio, "Multi-way, multilingual neural machine translation with a shared attention mechanism," in *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 2016, pp. 866–875.

[16] T.-L. Ha, J. Niehues, and A. Waibel, "Toward multilingual neural machine translation with universal encoder and decoder," *arXiv preprint arXiv:1611.04798*, 2016.

[17] N. Arivazhagan, A. Bapna, O. Firat, D. Lepikhin, M. Johnson, M. Krikun, M. X. Chen, Y. Cao, G. Foster, C. Cherry *et al.*, "Massively multilingual neural machine translation in the wild: Findings and challenges," *arXiv preprint arXiv:1907.05019*, 2019.

[18] B. Zoph, D. Yuret, J. May, and K. Knight, "Transfer learning for low-resource neural machine translation," in *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, 2016, pp. 1568–1575.

[19] O. Firat, B. Sankaran, Y. Al-Onaizan, F. T. Y. Vural, and K. Cho, "Zero-resource translation with multi-lingual neural machine translation," in *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, 2016, pp. 268–277.

[20] S. Sen, K. K. Gupta, A. Ekbal, and P. Bhattacharyya, "Multilingual unsupervised nmt using shared encoder and language-specific decoders," in *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, 2019, pp. 3083–3089.

[21] G. Blackwood, M. Ballesteros, and T. Ward, "Multilingual neural machine translation with task-specific attention," in *Proceedings of the 27th International Conference on Computational Linguistics*, 2018, pp. 3112–3122.

[22] T. Q. Nguyen and D. Chiang, "Transfer learning across low-resource, related languages for neural machine translation," in *Proceedings of the Eighth International Joint Conference on Natural Language Processing*, 2017, pp. 296–301.

[23] J. Gu, H. Hassan, J. Devlin, and V. O. Li, "Universal neural machine translation for extremely low resource languages," in *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 2018, pp. 344–354.

[24] D. He, Y. Xia, T. Qin, L. Wang, N. Yu, T. Liu, and W.-Y. Ma, "Dual learning for machine translation," in *Advances in neural information processing systems*, 2016.

[25] J. Zhang and C. Zong, "Exploiting source-side monolingual data in neural machine translation," in *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, 2016, pp. 1535–1545.

[26] A. Poncelas, D. Shterionov, A. Way, G. M. de Buy Wenniger, and P. Passban, "Investigating backtranslation in neural machine translation," in *Proceedings of the 21st Annual Conference of the European Association for Machine Translation: 28-30 May 2018, Universitat d'Alacant, Alacant, Spain*. European Association for Machine Translation, 2018, pp. 249–258.

[27] F. Burlot and F. Yvon, "Using monolingual data in neural machine translation: a systematic study," *arXiv preprint arXiv:1903.11437*, 2019.

[28] R. Cotterell and J. Kreutzer, "Explaining and generalizing back-translation through wake-sleep," *arXiv preprint arXiv:1806.04402*, 2018.

[29] S. Edunov, M. Ott, M. Auli, and D. Grangier, "Understanding back-translation at scale," in *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, 2018, pp. 489–500.

[30] H. Hassan, A. Aue, C. Chen, V. Chowdhary, J. Clark, C. Federmann, X. Huang, M. Junczys-Dowmunt, W. Lewis, M. Li *et al.*, "Achieving human parity on automatic chinese to english news translation," *arXiv preprint arXiv:1803.05567*, 2018.

[31] G. Lample, A. Conneau, L. Denoyer, and M. Ranzato, "Unsupervised machine translation using monolingual corpora only," in *International Conference on Learning Representations*, 2017.

[32] Y. Xia, X. Tan, F. Tian, T. Qin, N. Yu, and T.-Y. Liu, "Model-level dual learning," in *International Conference on Machine Learning*, 2018.

[33] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros, "Unpaired image-to-image translation using cycle-consistent adversarial networks," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 2223–2232.

[34] L. Wu, L. Zhao, T. Qin, J. Lai, and T.-Y. Liu, "Sequence prediction with unlabeled data by reward function learning." in *Proceedings of the 26th International Joint Conference on Artificial Intelligence*, 2017, pp. 3098–3104.

[35] C. Gulcehre, O. Firat, K. Xu, K. Cho, and Y. Bengio, "On integrating a language model into neural machine translation," *Computer Speech & Language*, vol. 45, pp. 137–148, 2017.

[36] P. Rajpurkar, J. Zhang, K. Lopyrev, and P. Liang, "Squad: 100,000+ questions for machine comprehension of text," in *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, 2016, pp. 2383–2392.

[37] S. Bowman, G. Angeli, C. Potts, and C. D. Manning, "A large annotated corpus for learning natural language inference," in *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, 2015, pp. 632–642.

[38] W. B. Dolan and C. Brockett, "Automatically constructing a corpus of sentential paraphrases," in *Proceedings of the Third International Workshop on Paraphrasing (IWP2005)*, 2005.

[39] A. Radford, K. Narasimhan, T. Salimans, and I. Sutskever, "Improving language understanding by generative pre-training," OpenAI, https://cdn.openai.com/research-covers/language-unsupervised/language_understanding_paper.pdf, Tech. Rep., 2018.

[40] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," in *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 2019, pp. 4171–4186.

[41] A. Conneau and G. Lample, "Cross-lingual language model pretraining," in *Advances in Neural Information Processing Systems*, 2019, pp. 7059–7069.

[42] M. Lewis, Y. Liu, N. Goyal, M. Ghazvininejad, A. Mohamed, O. Levy, V. Stoyanov, and L. Zettlemoyer, "Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension," *arXiv preprint arXiv:1910.13461*, 2019.

[43] Y. Liu, J. Gu, N. Goyal, X. Li, S. Edunov, M. Ghazvininejad, M. Lewis, and L. Zettlemoyer, "Multilingual denoising pre-training for neural machine translation," *arXiv preprint arXiv:2001.08210*, 2020.

[44] P. Vincent, H. Larochelle, Y. Bengio, and P.-A. Manzagol, "Extracting and composing robust features with denoising autoencoders," in *Proceedings of the 25th international conference on Machine learning*, 2008, pp. 1096–1103.

[45] G. Lample, M. Ott, A. Conneau, L. Denoyer, and M. Ranzato, "Phrase-based & neural unsupervised machine translation," in *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, 2018, pp. 5039–5049.

[46] M. Joshi, D. Chen, Y. Liu, D. S. Weld, L. Zettlemoyer, and O. Levy, "Spanbert: Improving pre-training by representing and predicting spans," *Transactions of the Association for Computational Linguistics*, vol. 8, pp. 64–77, 2020.

[47] B. Thompson, J. Gwinnup, H. Khayrallah, K. Duh, and P. Koehn, "Overcoming catastrophic forgetting during domain adaptation of neural machine translation," in *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 2019, pp. 2062–2068.

[48] Y. Xia, J. Bian, T. Qin, N. Yu, and T.-Y. Liu, "Dual inference for machine learning," in *Proceedings of the 26th International Joint Conference on Artificial Intelligence*, 2017.

[49] Z. Yi, H. R. Zhang, P. Tan, and M. Gong, "Dualgan: Unsupervised dual learning for -image translation." in *2017 IEEE International Conference on Computer Vision (ICCV)*, 2017, pp. 2868–2876.

[50] P. Luo, G. Wang, L. Lin, and X. Wang, "Deep dual learning for semantic image segmentation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017.

[51] N. Duan, D. Tang, P. Chen, and M. Zhou, "Question generation for question answering," in *Proceedings of the 2017 conference on empirical methods in natural language processing*, 2017, pp. 866–874.

[52] Y. Sun, D. Tang, N. Duan, S. Liu, Z. Yan, M. Zhou, Y. Lv, W. Yin, X. Feng, B. Qin *et al.*, "Joint learning of question answering and question generation," *IEEE Transactions on Knowledge and Data Engineering*, 2019.

[53] Q. Huang, P. Zhang, D. Wu, and L. Zhang, "Turbo learning for captionbot and drawingbot," in *Advances in neural information processing systems*, 2018.

[54] Z.-H. Zhou, *Ensemble methods: foundations and algorithms*. Chapman and Hall/CRC, 2012.

[55] G. Hinton, O. Vinyals, and J. Dean, "Distilling the knowledge in a neural network," in *NIPS Deep Learning and Representation Learning Workshop*, 2015.

[56] Y. Kim and A. M. Rush, "Sequence-level knowledge distillation," in *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, 2016.

[57] S. Edunov, M. Ott, M. Auli, D. Grangier, and M. Ranzato, "Classical structured prediction losses for sequence to sequence learning," in *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, 2018, pp. 355–364.

[58] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *International Conference on Learning Representations (ICLR)*, 2015.

[59] K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu, "Bleu: a method for automatic evaluation of machine translation," in *Proceedings of the 40th annual meeting on association for computational linguistics*, 2002.

[60] M. Junczys-Dowmunt, "Microsoft's submission to the wmt2018 news translation task: How i learned to stop worrying and love the data," in *Proceedings of the Third Conference on Machine Translation: Shared Task Papers*, 2018, pp. 425–430.

[61] D. Ciregan, U. Meier, and J. Schmidhuber, "Multi-column deep neural networks for image classification," in *2012 IEEE conference on computer vision and pattern recognition*. IEEE, 2012, pp. 3642–3649.

[62] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *Advances in neural information processing systems*, 2014.

[63] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros, "Image-to-image translation with conditional adversarial networks," in *Computer Vision and Pattern Recognition (CVPR), 2017 IEEE Conference on*. IEEE, 2017.

[64] M. Heusel, H. Ramsauer, T. Unterthiner, B. Nessler, and S. Hochreiter, "Gans trained by a two time-scale update rule converge to a local nash equilibrium," in *BIPS*, 2017.

[65] M. Lucic, K. Kurach, M. Michalski, S. Gelly, and O. Bousquet, "Are gans created equal? a large-scale study," in *Advances in neural information processing systems*, 2018.

[66] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015.

[67] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele, "The cityscapes dataset for semantic urban scene understanding," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016.

[68] J. Gu, J. Bradbury, C. Xiong, V. Li, and R. Socher, "Non-autoregressive neural machine translation," in *International Conference on Learning Representations (ICLR)*, 2018.

[69] Y. Wang, F. Tian, D. He, T. Qin, Z. ChengXiang, and T.-Y. Liu, "Non-autoregressive machine translation with auxiliary regularization," in *Thirty-Third AAAI Conference on Artificial Intelligence*, 2019.

[70] J. Guo, T. Xu, D. He, T. Qin, L. Xu, and T.-Y. Liu, "Non-autoregressive neural machine translation with enhanced decoder input," in *Thirty-Third AAAI Conference on Artificial Intelligence*, 2019.

[71] Y. Wang, Y. Xia, L. Zhao, J. Bian, T. Qin, G. Liu, and T. Liu, "Dual transfer learning for neural machine translation with marginal distribution regularization," in *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

[72] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," in *International Conference on Learning Representations*, 2015.

[73] C. Gulcehre, O. Firat, K. Xu, K. Cho, L. Barrault, H.-C. Lin, F. Bougares, H. Schwenk, and Y. Bengio, "On using monolingual corpora in neural machine translation," *arXiv preprint arXiv:1503.03535*, 2015.

[74] Z. Zhang, S. Liu, M. Li, M. Zhou, and E. Chen, "Joint training for neural machine translation models with monolingual data," in *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

[75] Y. Cheng, W. Xu, Z. He, W. He, H. Wu, M. Sun, and Y. Liu, "Semi-supervised learning for neural machine translation," in *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, vol. 1, 2016, pp. 1965–1974.

[76] M. Chinea-Rios, A. Peris, and F. Casacuberta, "Adapting neural machine translation with parallel synthetic data," in *Proceedings of the Second Conference on Machine Translation*, 2017, pp. 138–147.

[77] L. Wu, F. Tian, T. Qin, J. Lai, and T.-Y. Liu, "A study of reinforcement learning for neural machine translation," in *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, 2018, pp. 3612–3621.

[78] N. Ueffing, G. Haffari, and A. Sarkar, "Transductive learning for statistical machine translation," in *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, 2007, pp. 25–32.

[79] M.-T. Luong, H. Pham, and C. D. Manning, "Effective approaches to attention-based neural machine translation," in *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, 2015, pp. 1412–1421.

[80] J. Gu, Z. Lu, H. Li, and V. O. Li, "Incorporating copying mechanism in sequence-to-sequence learning," in *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, vol. 1, 2016, pp. 1631–1640.

[81] Y. Wang, Y. Xia, T. He, F. Tian, T. Qin, C. Zhai, and T.-Y. Liu, "Multi-agent dual learning," in *International Conference on Learning Representations*, 2019.

[82] M. Artetxe, G. Labaka, E. Agirre, and K. Cho, "Unsupervised neural machine translation," in *International Conference on Learning Representations*, 2017.

[83] Z. Xie, S. I. Wang, J. Li, D. Lévy, A. Nie, D. Jurafsky, and A. Y. Ng, "Data noising as smoothing in neural network language models," in *International Conference on Learning Representations*, 2017.

[84] S. Kobayashi, "Contextual augmentation: Data augmentation by words with paradigmatic relations," in *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, 2018, pp. 452–457.

[85] J. Zhu, F. Gao, L. Wu, X. Chen, T. Qin, and T.-Y. Liu, "Soft contextual data augmentation for neural machine translation," in *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, 2019, pp. 5539–5544.

[86] M. Lui and T. Baldwin, "langid. py: An off-the-shelf language identification tool," in *Proceedings of the ACL 2012 system demonstrations*. Association for Computational Linguistics, 2012, pp. 25–30.

[87] P. Koehn, H. Hoang, A. Birch, C. Callison-Burch, M. Federico, N. Bertoldi, B. Cowan, W. Shen, C. Moran, R. Zens *et al.*, "Moses: Open source toolkit for statistical machine translation," in *Proceedings of the 45th annual meeting of the association for computational linguistics companion volume proceedings of the demo and poster sessions*, 2007, pp. 177–180.

[88] R. Sennrich, B. Haddow, and A. Birch, "Neural machine translation of rare words with subword units," in *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, vol. 1, 2016, pp. 1715–1725.

[89] M. Ott, S. Edunov, A. Baevski, A. Fan, S. Gross, N. Ng, D. Grangier, and M. Auli, "fairseq: A fast, extensible toolkit for sequence modeling," in *Proceedings of NAACL-HLT 2019: Demonstrations*, 2019.

[90] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[91] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 2818–2826.

[92] M. Post, "A call for clarity in reporting bleu scores," in *Proceedings of the Third Conference on Machine Translation: Research Papers*, 2018, pp. 186–191.

[93] M. Graça, Y. Kim, J. Schamper, J. Geng, and H. Ney, "The rwth aachen university english-german and german-english unsupervised neural machine translation systems for wmt 2018," in *Proceedings of the Third Conference on Machine Translation: Shared Task Papers*, 2018, pp. 377–385.

[94] F. Stahlberg, A. de Gispert, and B. Byrne, "The university of cambridge's machine translation systems for wmt18," in *Proceedings of the Third Conference on Machine Translation: Shared Task Papers*, 2018, pp. 504–512.

[95] L. Wu, Y. Wang, Y. Xia, Q. Tao, J. Lai, and T.-Y. Liu, "Exploiting monolingual data at scale for neural machine translation," in *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, 2019, pp. 4198–4207.

[96] R. Caruana, "Multitask learning," *Machine learning*, pp. 41–75, 1997.

[97] R. Collobert and J. Weston, "A unified architecture for natural language processing: Deep neural networks with multitask learning," in *Proceedings of the 25th international conference on Machine learning*, 2008, pp. 160–167.

[98] L. Deng, G. Hinton, and B. Kingsbury, "New types of deep neural network learning for speech recognition and related applications: An overview," in *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*, 2013.

[99] S. Ruder, "An overview of multi-task learning in deep neural networks," *arXiv preprint arXiv:1706.05098*, 2017.

[100] M.-T. Luong, Q. V. Le, I. Sutskever, O. Vinyals, and L. Kaiser, "Multi-task sequence to sequence learning," *arXiv preprint arXiv:1511.06114*, 2015.

[101] J. Niehues and E. Cho, "Exploiting linguistic resources for neural machine translation using multi-task learning," in *Proceedings of the Second Conference on Machine Translation*, 2017, pp. 80–89.

[102] A. Eriguchi, Y. Tsuruoka, and K. Cho, "Learning to parse and translate improves neural machine translation," in *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics*, 2017, pp. 72–78.

[103] P. Zaremoodi and G. Haffari, "Neural machine translation for bilingually scarce scenarios: a deep multi-task learning approach," in *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 2018.

[104] E. Kiperwasser and M. Ballesteros, "Scheduled multi-task learning: From syntax to translation," *Transactions of the Association for Computational Linguistics*, pp. 225–240, 2018.

[105] A. Bapna and O. Firat, "Simple, scalable adaptation for neural machine translation," in *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*, 2019, pp. 1538–1548.

[106] J. L. Elman, "Learning and development in neural networks: The importance of starting small," *Cognition*, pp. 71–99, 1993.

[107] T. Kudo and J. Richardson, "Sentencepiece: A simple and language independent subword tokenizer and detokenizer for neural text processing," *arXiv preprint arXiv:1808.06226*, 2018.

[108] G. Wenzek, M.-A. Lachaux, A. Conneau, V. Chaudhary, F. Guzman, A. Joulin, and E. Grave, "Ccnet: Extracting high quality monolingual datasets from web crawl data," *arXiv preprint arXiv:1911.00359*, 2019.

[109] A. De Gispert and J. B. Marino, "Catalan-english statistical machine translation without parallel corpus: bridging through spanish," in *Proc. of 5th International Conference on Language Resources and Evaluation (LREC)*. Citeseer, 2006, pp. 65–68.

[110] M. Utiyama and H. Isahara, "A comparison of pivot methods for phrase-based statistical machine translation," in *Human Language Technologies 2007: The Conference of the North American Chapter of the Association for Computational Linguistics; Proceedings of the Main Conference*, 2007, pp. 484–491.

[111] A. Conneau, K. Khandelwal, N. Goyal, V. Chaudhary, G. Wenzek, F. Guzmán, E. Grave, M. Ott, L. Zettlemoyer, and V. Stoyanov, "Unsupervised cross-lingual representation learning at scale," in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, D. Jurafsky, J. Chai, N. Schluter, and J. R. Tetreault, Eds. Association for Computational Linguistics, 2020, pp. 8440–8451. [Online]. Available: https://www.aclweb.org/anthology/2020.acl-main.747/

[112] A. Conneau, R. Rinott, G. Lample, A. Williams, S. Bowman, H. Schwenk, and V. Stoyanov, "XNLI: Evaluating cross-lingual sentence representations," in *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*. Brussels, Belgium: Association for Computational Linguistics, Oct.-Nov. 2018. [Online]. Available: https://www.aclweb.org/anthology/D18-1269

[113] Y. Liang, N. Duan, Y. Gong, N. Wu, F. Guo, W. Qi, M. Gong, L. Shou, D. Jiang, G. Cao, X. Fan, R. Zhang, R. Agrawal, E. Cui, S. Wei, T. Bharti, Y. Qiao, J.-H. Chen, W. Wu, S. Liu, F. Yang, D. Campos, R. Majumder, and M. Zhou, "Xglue: A new benchmark dataset for cross-lingual pre-training, understanding and generation," *arXiv*, vol. abs/2004.01401, 2020.

[114] A. Siddhant, M. Johnson, H. Tsai, N. Ari, J. Riesa, A. Bapna, O. Firat, and K. Raman, "Evaluating the cross-lingual effectiveness of massively multilingual neural machine translation," in *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020*. AAAI Press, 2020, pp. 8854–8861. [Online]. Available: https://aaai.org/ojs/index.php/AAAI/article/view/6414

[115] T. Brown, B. Mann, N. Ryder, M. Subbiah, J. D. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell *et al.*, "Language models are few-shot learners," *Advances in neural information processing systems*, vol. 33, pp. 1877–1901, 2020.

[116] OpenAI, "Gpt-4 technical report," *arXiv preprint arXiv:2303.08774*, 2023.

[117] A. Chowdhery, S. Narang, J. Devlin, M. Bosma, G. Mishra, A. Roberts, P. Barham, H. W. Chung, C. Sutton, S. Gehrmann *et al.*, "Palm: Scaling language modeling with pathways," *arXiv preprint arXiv:2204.02311*, 2022.

[118] R. Sennrich, B. Haddow, and A. Birch, "Controlling politeness in neural machine translation via side constraints," in *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 2016, pp. 35–40.

[119] A. Viswanathan, V. Wang, and A. Kononova, "Controlling formality and style of machine translation output using automl," in *Annual International Symposium on Information Management and Big Data*. Springer, 2019, pp. 306–313.

[120] A. Schioppa, D. Vilar, A. Sokolov, and K. Filippova, "Controlling machine translation for multiple attributes with additive interventions," in *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, 2021, pp. 6676–6696.