Point, Disambiguate and Copy: Incorporating Bilingual Dictionaries for Neural Machine Translation

Tong Zhang[1,2], **Long Zhang**[1,2], **Wei Ye**[1,†], **Bo Li**[1,2],
Jinan Sun[1], **Xiaoyu Zhu**[3], **Wen Zhao**[1], **Shikun Zhang**[1,†]
[1] National Engineering Research Center for Software Engineering, Peking University
[2] School of Software and Microelectronics, Peking University
[3] BIGO
{zhangtong17,zhanglong418, wye, zhangsk}@pku.edu.cn

Abstract

This paper proposes a sophisticated neural architecture to incorporate bilingual dictionaries into Neural Machine Translation (NMT) models. By introducing three novel components: ***P**ointer*, ***D**isambiguator*, and ***C**opier*, our method PDC achieves the following merits inherently compared with previous efforts: (1) *Pointer* leverages the semantic information from bilingual dictionaries, for the first time, to better locate source words whose translation in dictionaries can potentially be used; (2) *Disambiguator* synthesizes contextual information from the source view and the target view, both of which contribute to distinguishing the proper translation of a specific source word from multiple candidates in dictionaries; (3) *Copier* systematically connects *Pointer* and *Disambiguator* based on a hierarchical copy mechanism seamlessly integrated with Transformer, thereby building an end-to-end architecture that could avoid error propagation problems in alternative pipeline methods. The experimental results on Chinese-English and English-Japanese benchmarks demonstrate the PDC's overall superiority and effectiveness of each component.

1 Introduction

The past several years have witnessed the remarkable success of Neural machine translation (NMT), due to the development of sequence-to-sequence methods (Sutskever et al., 2014; Bahdanau et al., 2015; Vaswani et al., 2017). Since bilingual dictionaries cover rich prior knowledge, especially of low-frequency words, many efforts have been dedicated to incorporating bilingual dictionaries into NMT systems. These explorations can be roughly categorized into two broad paradigms. The first one transforms the bilingual dictionaries into pseudo parallel sentence pairs for training (Zhang and Zong, 2016; Zhao et al., 2020). The second one utilizes the bilingual dictionaries as external resources fed into neural architectures (Luong et al., 2015; Gulcehre et al., 2016; Arthur et al., 2016; Zhang et al., 2017b; Zhao et al., 2018a,b, 2019b), which is more widely used and the focus of this paper.

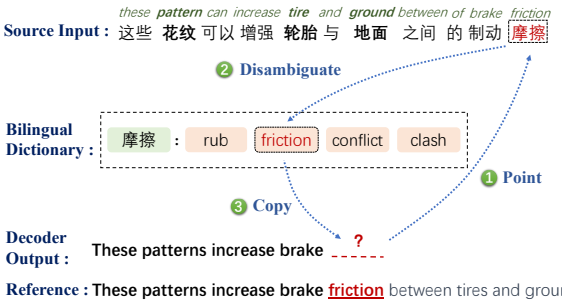

Figure 1: Three key steps to translate with a bilingual dictionary: pointing, disambiguating and copying. This concrete illustrative example is chosen to conveniently show the primary intuition behind our method.

In practice, bilingual dictionaries usually contain more than one translation for a word. From a high-level perspective, we believe there are three critical steps to incorporate bilingual dictionaries into NMT models as shown in Figure 1: (1) *pointing* to a source word whose translation in dictionaries will be used at a decoding step, (2) *disambiguating* multiple translation candidates of the source word from dictionaries, and (3) *copying* the selected translation into the target side if necessary. Note that some works assume that only one translation exists for each word in dictionaries (Luong et al., 2015; Gulcehre et al., 2016). In this simplified scenario, the disambiguating step is unnecessary, hence the pointing and copying step can be merged into a single step similar to the classic copying mechanism (Gu et al., 2016). In more practical scenarios, however, this process suffers from the following bottlenecks corresponding to each step.

[†]Corresponding authors.

Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, pages 3970–3979
August 1–6, 2021.

(1) In the *pointing* step, semantic information of translations in dictionaries is underutilized. To locate source words whose translation in dictionaries may be used, some works (Luong et al., 2015; Gulcehre et al., 2016) use a classic copy mechanism, but in an oversimplified scenario mentioned above. More recent efforts further leverage statistics-based pre-processing methods (Zhao et al., 2018b, 2019b) to help identify, e.g., rare or troublesome source words. Note that the goal of locating a source word is to further use its translation in dictionaries. Intuitively, by exploring rich information of a source word's translations in dictionaries, we can better understand the semantic meaning of the source word and distinguish whether we can its translation candidate. Unfortunately, this information is underutilized by most works, which could have boosted NMT performance, as shown in Section 5.2.

(2) In the *disambiguating* step, the distinguishing information is from static prior knowledge or coarse-grained context information. To select the proper translation of one source word from multiple candidates in dictionaries, in addition to works that merely use the first-rank one (Luong et al., 2015; Gulcehre et al., 2016), existing explorations mainly involve exploiting prior probabilities, e.g., to adjust the distribution over the decoding vocabulary (Arthur et al., 2016; Zhao et al., 2018a). As a representative context-based disambiguation method, Zhao et al. (2019b) distinguish candidates by matching their embeddings with a decoder-oriented context embedding. Intuitively, an optimal translation candidate should not only accurately reflect the content of the source sentence, but also be consistent with the context of the current partial target sentence. Our observation is that both source information and target information is critical and complementary to distinguish candidates. Taking the source word "摩擦" in Figure 1 for example, the source context of "花纹/*pattern*", "轮胎/*tire*" and "地面/*ground*" helps to identify the candidates of "*rub*" and "*friction*" in the dictionary, and the target context of "*these patterns increase brake*" further makes "*friction*" the best choice. This observation inspires us to synthesize source information and target information in a more fine-grained manner to improve previous straightforward disambiguation methods.

(3) A *copying* step is required to facilitate the collaboration between the pointing step and disambiguating step. Existing models usually do not explicitly emphasize a separate copying step [1], since it is a trivial task in their simplified or pipeline scenario. However, to deliver a sophisticated end-to-end architecture that avoids error propagation problems, the pointing and disambiguating step must be appropriately connected as well as integrated into mature NMT models. The proposed copying step is the right place to complete this job.

To address the above problems, we propose a novel neural architecture consisting of three novel components: ***P**ointer*, ***D**isambiguator*, and ***C**opier*, to effectively incorporate bilingual dictionaries into NMT models in an end-to-end manner. *Pointer* is a pioneering research effort on exploiting the semantic information from bilingual dictionaries to better locate source words whose translation in dictionaries may be used. *Disambiguator* synthesizes complementary contextual information from the source and target via a bi-view disambiguation mechanism, accurately distinguishing the proper translation of a specific source word from multiple candidates in dictionaries. *Copier* couples *Pointer* and *Disambiguator* based on a hierarchical copy mechanism seamlessly integrated with Transformer, thereby building a sophisticated end-to-end architecture. Last but not least, we design a simple and effective method to integrate byte-pair encoding (BPE) with bilingual dictionaries in our architecture. Extensive experiments are performed on Chinese-English and English-Japanese benchmarks, and the results verify the PDC's overall performance and effectiveness of each component.

2 Background: Transformer

Transformer (Vaswani et al., 2017) is the most popular NMT architecture, which adopts the standard encoder-decoder framework and relies solely on stacked attention mechanisms. Specifically, given a source sequence $\mathbf{x} = \{x_1, x_2..., x_n\}$, the model is supposed to generate the target sequence $\mathbf{y} = \{y_1, y_2..., y_m\}$ in an auto-regressive paradigm.

Transformer Encoder. A Transformer encoder is constituted by a stack of N identical layers, each of which contains two sub-layers. The first is a multi-head self-attention mechanism (SelfAtt), and the second is a fully connected feed-forward network (FFN). Layer normalization (LN) (Ba et al., 2016) and residual connection (He et al., 2016) is em-

[1] Note that previous works involve copy mechanism mainly correspond to the Pointing step.

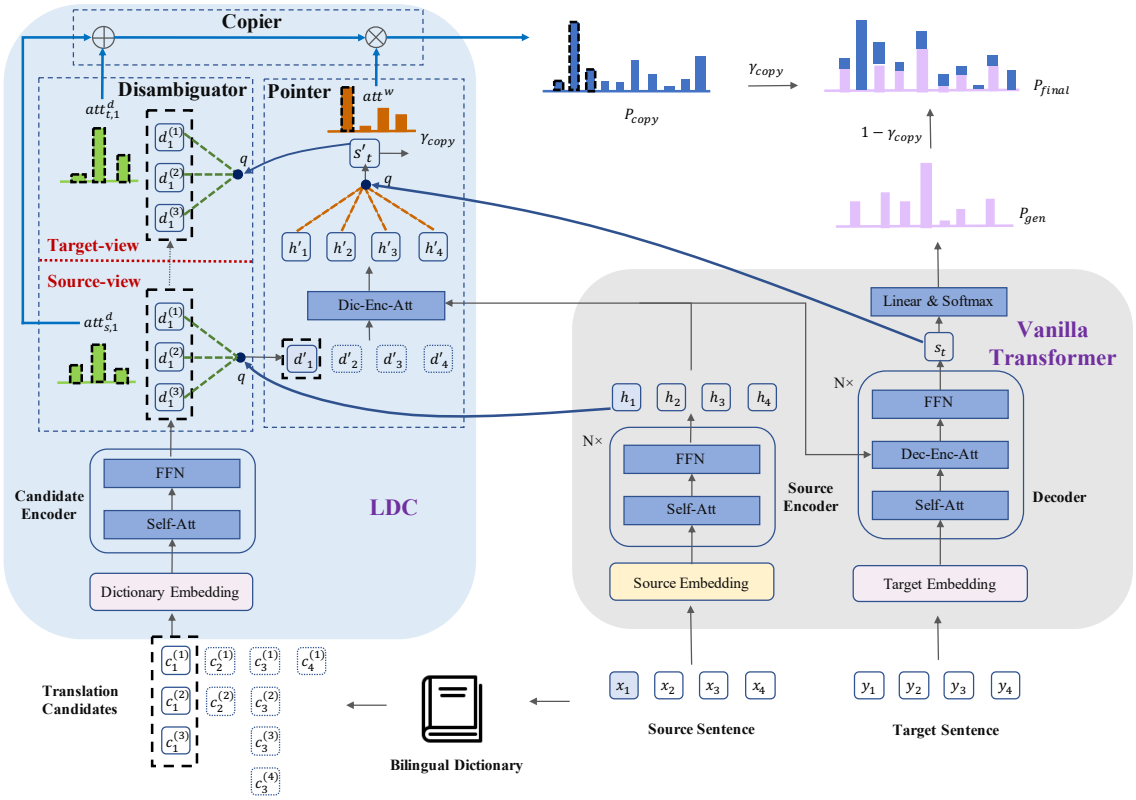

Figure 2: An overview of our methods. The left is our PDC module as a copy mechanism, and the right is the vanilla Transformer. For each source word x_i, we obtain a set of translation candidates $\{c_i^{(1)}, ..., c_i^{(k)}\}$ via a bilingual dictionary. To better capture their semantics, candidate embeddings are shared with target embeddings and refined with self-attention before interacting with Transformer's encoder states. The state h' enriched by candidate semantics is utilized by *Pointer* to locate source words whose dictionary translations may be used. *Disambiguator* generates two disambiguation distributions over translation candidates from the source view and target view, respectively. Finally, *Copier* connects the outputs of *Pointer* and *Disambiguator* via a hierarchical copy operation.

ployed around the two sub-layers in both encoder and decoder.

$$\begin{aligned}\tilde{\mathbf{h}}^l &= \mathrm{LN}(\mathrm{SelfAtt}(\mathbf{h}^{l-1}) + \mathbf{h}^{l-1}),\\ \mathbf{h}^l &= \mathrm{LN}(\mathrm{FFN}(\tilde{\mathbf{h}}^l) + \tilde{\mathbf{h}}^l),\end{aligned} \tag{1}$$

where $\mathbf{h}^l = \{h_1^l, h_2^l ..., h_n^l\}$ is the output of the l-th layer. The final output $\mathbf{h}^N$ of the last encoder layer serves as the encoder state $\mathbf{h}$.

Transformer Decoder. Similarly, the decoder employs the stack structure with N layers. Besides the two sub-layers, an additional cross attention (CrossAtt) sub-layer is inserted to capture the information from the encoder.

$$\begin{aligned}\tilde{\mathbf{s}}^l &= \mathrm{LN}(\mathrm{SelfAtt}(\mathbf{s}^{l-1}) + \mathbf{s}^{l-1}),\\ \widehat{\mathbf{s}}^l &= \mathrm{LN}(\mathrm{CrossAtt}(\tilde{\mathbf{s}}^l, \mathbf{h}, \mathbf{h}) + \tilde{\mathbf{s}}^l),\\ \mathbf{s}^l &= \mathrm{LN}(\mathrm{FFN}(\widehat{\mathbf{s}}^l) + \widehat{\mathbf{s}}^l),\end{aligned} \tag{2}$$

where $\mathbf{s}^l$ is the output of the l-th decoder layer and the final output $\mathbf{s}^N$ is taken as the decoder state $\mathbf{s}$.

Then, the translation probability $p(y_t|\mathbf{y}_{<t}, \mathbf{x})$ of the t-th target word is produced with a softmax layer:

$$p(y_t|\mathbf{y}_{<t}, \mathbf{x}) \propto \exp(\mathbf{W}_o s_t), \tag{3}$$

where $\mathbf{y}_{<t}$ is the proceeding tokens before y_t.

3 Methodology

In this section, we mathematically describe our model in detail. We follow the notations in Section 2. $\mathbf{c_i} = \{c_i^{(1)}, ..., c_i^{(k)}\}$ denotes the translation candidates of a source word x_i, derived from a bilingual dictionary $\mathbf{D}$.

3.1 Overview

An overview of the proposed PDC model is shown in Figure 2. PDC aims to copy the correct translation candidate of the correct source word at a decoding step. Following the classic CopyNet (Gu et al., 2016), our model consists of two parts, an

encoder-decoder translator to produce the generating probability and a copy mechanism to produce the copying probability. The above two probabilities will collaborate to emit the final probability.

The procedure of our copy mechanism involves three critical components: (1) a ***Pointer*** that selects a source word whose translation candidates will potentially be copied, (2) a ***Disambiguator*** which distinguishes multiple translation candidates of the source word to find the optimal candidate to copy, and (3) a ***Copier*** that generates copying probability by combining the outputs from the above two components hierarchically. We will describe the details of each component in the following subsection.

3.2 Pointer

The pointer aims to point which source word should be translated at a decoding step. We utilize the carefully extracted semantic information of translation candidates to promote pointing accuracy. Specifically, pointer first extracts the semantic information of candidates with candidate-wise encoding. Then the candidate representations of each source word are fused and interacted with the source representations from transformer encoder. An attention mechanism is applied on the refined source representations to point which word to be translated.

Candidate Encoding. We first construct the candidate representations $\mathbf{d}_i = \{d_i^{(1)}, ..., d_i^{(k)}\}$ for the candidates of x_i, through an candidate embedding matrix and a single layer candidate encoder.

$$\begin{aligned}\tilde{\mathbf{d}}_i &= \mathrm{LN}(\mathrm{SelfAtt}(\mathrm{Emb}(\mathbf{c}_i)) + \mathrm{Emb}(\mathbf{c}_i)),\\ \mathbf{d}_i &= \mathrm{LN}(\mathrm{FFN}(\tilde{\mathbf{d}}_i) + \tilde{\mathbf{d}}_i).\end{aligned} \tag{4}$$

Note that this candidate-wise encoder exploits the same structure as a source encoder layer.

Pointing with candidate semantics. Previous dictionary-enhanced NMT systems usually directly utilize encoder state $\mathbf{h}$ and the decoder state s_t at t-th decoding step to point whose translation should be copied in the source sentence. Intuitively, translation candidates' information contributes to pointing the right source word, while it is underutilized previously. Accordingly, we propose to explore the semantic information of translation candidates in our pointer. First, we fuse multiple translation candidates' representations of each word by an attention mechanism between h_i and $\mathbf{d}_i$.

$$d'_i = \sum_{j=1}^{k} \alpha_{i,j}^{\mathrm{src}} \cdot d_i^{(j)}; \alpha_{i,j}^{\mathrm{src}} = \frac{\exp(h_i \mathbf{W} d_i^{(j)})}{\sum_{j'=1}^{k} \exp(h_i \mathbf{W} d_i^{(j')})}, \tag{5}$$

where $d'_i \in \mathbf{d}'$ is the fused representation for all candidates of the source word x_i. Next, the encoder state $\mathbf{h}$ and $\mathbf{d}'$ are interacted to refine the representations of source words with the carefully-extracted candidate information. The refined encoder state $\mathbf{h}'$ can be formalized as:

$$\begin{aligned}\tilde{\mathbf{h}'} &= \mathrm{LN}(\mathrm{CrossAtt}(\mathbf{h}', \mathbf{d}', \mathbf{d}') + \mathbf{h}'),\\ \mathbf{h}' &= \mathrm{LN}(\mathrm{FFN}(\tilde{\mathbf{h}'}) + \tilde{\mathbf{h}'}).\end{aligned} \tag{6}$$

Then, we calculate the attention score to point which source word to be translated:

$$s'_t = \sum_{i=1}^{n} \beta_i \cdot h'_i;\ \beta_i = \frac{\exp(s_t \mathbf{W} h'_i)}{\sum_{i'=1}^{n} \exp(s_t \mathbf{W} h'_{i'})}, \tag{7}$$

where β_i is the pointing probability for x_i. s'_t denotes the refined decoder state.

3.3 Disambiguator

When translating a specific word, our model has the whole source sentence and the partial target sentence as inputs. An optimal translation candidate should not only accurately reflect the content of source sentence, but also be consistent with the context of the partial target sentence. Thus, we propose a bi-view disambiguation module to select the optimal translation candidate in both source view and target view.

Source-view Disambiguation. Source-view disambiguation chooses the optimal candidate for each word with the context information stored in source sentence. The attention score $\boldsymbol{\alpha}_i^{\mathrm{src}} = \{\alpha_{i,1}^{\mathrm{src}}, ..., \alpha_{i,k}^{\mathrm{src}}\}$, which has been calculated in Equation 5, is employed as the source-view disambiguating distribution for the k translation candidates of x_i. This disambiguating distribution is decoding-agnostic, which means it serve as global information during decoding.

Target-view Diambiguation. As analyzed in Section 1, translation candidates that seem proper from the source view may not well fit in the target context. Thus, we also perform a target view disambiguation to narrow down which candidates fit the partial target sentence's context. Specifically, we leverage the refined decoder state s'_t to disambiguate the multiple candidates:

$$\alpha_{i,j}^{\mathrm{tgt}} = \frac{\exp(s'_t \mathbf{W}_{dt} d_i^{(j)})}{\sum_{j'=1}^{k} \exp(s'_t \mathbf{W}_{dt} d_i^{(j')})}, \tag{8}$$

where $\alpha_{i,j}^{\mathrm{tgt}}$ is the target-view disambiguating probability for $c_i^{(j)}$. In contrast to the decoding-agnostic

source-view disambiguating probability, this target-view disambiguating probability varies during decoding steps.

3.4 Copier

Finally, we combine the pointing distribution and the bi-view disambiguating distributions in a hierarchical way to constitute the copying distribution as follows:

$$\alpha_{i,j} = [\rho \times \alpha^{\text{src}}_{i,j} + (1-\rho) \times \alpha^{\text{tgt}}_{i,j}] \times \beta_i, \quad (9)$$

where ρ is a scaling factor to adjust the contribution from source-view and target-view disambiguating probabilities. $\alpha_{i,j}$ indicates the probability to copy $c_i^{(j)}$, the j-th translation candidate of the i-th source word. We transform this positional probability into word-level copying probability p_{copy}:

$$p_{\text{copy}} = p(y_t|\mathbf{y}_{<t}, \mathbf{x}, \mathbf{c}), \quad (10)$$

where $\mathbf{c}$ is the entire translation candidates for all source word in an instance.

The final probability p_{final} is constituted by a linear interpolation of p_{gen} and p_{copy}:

$$p_{\text{final}}(y_t|\mathbf{y}_{<t}, \mathbf{x}, \mathbf{c}) = \gamma_t \times p_{\text{copy}} + (1-\gamma_t) \times p_{\text{gen}}, \quad (11)$$

where p_{gen} denotes the the generating probability from Transformer, calculated in Equation 3. γ_t is the dynamic weight at step t, formalized by:

$$\gamma_t = \text{sigmoid}(\mathbf{W}\text{s}'_t). \quad (12)$$

3.5 Selective BPE

BPE (Sennrich et al., 2016) is commonly used in NMT to deal with the rare words by separating them into frequent subwords. However, it is non-trivial to incorporate BPE into NMT systems with copy mechanism, because the split subwords may not match the original word appearing in dictionaries, either in source side or target side. Simply applying BPE on dictionary words will complicates the scenario to disambiguate and copy, since the model needs to aggregate the representations of these subwords for disambiguation and copy the subwords sequentially. As revealed in Section 5.4, the experimental results demonstrate that whether applying original BPE on dictionary words or not will not yield promising results.

In this paper, we present a simple and effective strategy named ***selective BPE***, which only performs BPE on all source words and a portion of target words. All of the translation candidates from the dictionary remain intact. Concretely, in the target side, we keep the target word from being separated into subwords if we can copy it from the translation candidate set $\mathbf{c}$ of the source sentence. Such case is formalized as:

$$I_{\text{tgt}}(i) = \begin{cases} 1, & \text{if } y_i \in \mathbf{c} \\ 0, & \text{if } y_i \notin \mathbf{c} \end{cases}, \quad (13)$$

where $I_{\text{tgt}}(i)$ is the BPE indicator for y_i. A target word y_i will be split by selective BPE only if $I_{\text{tgt}}(i) = 0$. Note that selective BPE is only used in training, since the reference of validation sets and testing sets do not need BPE.

By applying selective BPE, our model can implicitly exploit the information of which dictionary candidates are likely to be copied. Thus, rare words will be more inclined to be copied directly as a whole from the dictionary.

4 Experimental Settings

In this section, we elaborate on the experiment setup to evaluate our proposed model.

4.1 Datasets

We test our model on Chinese-to-Engish (Zh-En) and English-Japanese (En-Ja) translation tasks.

For Zh-En translation, we carry out experiments on two datesets. We use 1.25M sentence pairs from news corpora LDC as the training set [1]. We adopt NIST 2006 (MT06) as validation set. NIST 2002, 2003, 2004, 2005, 2008 datasets are used for testing. Besides, we use the Ted talks corpus from IWSLT 2014 and 2015 (Cettolo et al., 2012) including 0.22M sentence pairs for training. We use *dev2010* with 0.9K sentence pairs for development and *tst2010-2013* with 5.5K sentence pairs for testing.

For En-Ja translation, we adopt Wikipedia article dataset KFTT[2], which contains 0.44M sentence pairs for training, 1.2K sentence pairs for validation and 1.2K sentence pairs for testing.

The bilingual dictionary we used is constructed by the open-source cross-lingual word translate dataset word2word (Choe et al., 2020). We limit the maximum number of translation candidates to 5 for each source word.

[1]The training set includes LDC2002E18, LDC2003E07, LDC2003E14, Hansards portion of LDC2004T07, LDC2004T08 and LDC2005T06.

[2]http://www.phontron.com/kftt/

Systems	MT06	MT02	MT03	MT04	MT05	MT08	Δ
Exsisting NMT systems							
(Cheng et al., 2019)	46.95	47.06	46.48	47.39	46.58	37.38	-
(Yang et al., 2020)	44.69	-	46.56	-	46.04	37.53	-
(Yan et al., 2020)	47.80	47.72	46.60	48.30	-	**38.70**	-
Baseline NMT systems							
Transformer	44.11	46.38	45.05	47.07	44.82	34.74	ref
Single-Copy	45.04	47.21	46.47	47.48	45.45	36.08	+0.93
Flat-Copy	44.93	46.33	46.26	46.83	45.38	35.19	+0.39
Our NMT systems							
PDC	46.74	**48.85**	**48.43**	**48.57**	**47.71**	37.45	**+2.59**
PDC(w/o Dict-Pointer)	45.79	47.58	47.81	47.98	46.32	36.53	+1.63
PDC(w/o Tgt-View)	45.80	47.43	47.91	48.49	46.81	36.99	+1.91
PDC(w/o Src-View)	45.97	47.42	47.90	47.92	47.07	36.81	+1.81

Table 1: The main results of NIST Zh-En task. Δ shows the average BLEU improvements over the test sets compared with Transformer (ref). The results of our models significantly outperform Transformer ($p < 0.01$).

4.2 Details for Training and Evaluation

We implement our model on top of THUMT (Zhang et al., 2017a) toolkit. The dropout rate is set to be 0.1. The size of a mini-batch is 4096. We share the parameters in target embeddings and the output matrix of the Transformer decoder. The other hyper-parameters are the same as the default settings in Vaswani et al. (2017). The optimal value scaling factor ρ in bi-view disambiguation is 0.4. All these hyper-parameters are tuned on the validation set. We apply BPE (Sennrich et al., 2016) with 32K merge operations. The best single model in validation is used for testing. We use $multi{-}bleu.perl$[3] to calculate the case-insensitive 4-gram BLEU.

4.3 Baselines

Our models and the baselines use BPE in experiments by default. We compare our PDC with the following baselines:

- **Transformer** is the most widely-used NMT system with self-attention (Vaswani et al., 2017).
- **Single-Copy** is a Transformer-based copy mechanism that select a source word's first-rank translation candidate exactly following Luong et al. (2015); Gulcehre et al. (2016).
- **Flat-Copy** is a novel copy mechanism to perform automatic post-editing (APE) proposed by Huang et al. (2019). Note that APE focuses on copying from a draft generated by a pre-trained NMT system. We first arrange candidates of all source words into a sequence as a draft and then copy this flattened "draft" following Huang et al. (2019).

[3]https://github.com/moses-smt/mosesdecoder/blob/master/scripts/generic/multi-bleu.perl

5 Experiment Results

5.1 Main Results

Table 1 shows the performance of the baseline models and our method variants. We also list several existing robust NMT systems reported in previous work to validate PDC's effectiveness. By investigating the results in Table 1, we have the following four observations.

First, compared with existing state-of-the-art NMT systems, PDC achieves very competitive results, e.g., the best BLEU scores in 4 of the 5 test sets.

Second, Single-Copy outperforms Transformer, indicating that even incorporating only the first-rank translation candidate can improve NMT models. However, since Single-Copy disregards many translation candidates in dictionaries, which could have been copied, the improvement is relatively small (e.g., +0.93 of average BLEU score on the test sets).

Third, the performance of Flat-Copy is even worse than Single-Copy, though it considers all translation candidates in dictionaries. The reason lies in that Flat-Copy ignores the hierarchy formed by a source sentence and the corresponding translation candidates of its each word, making it much

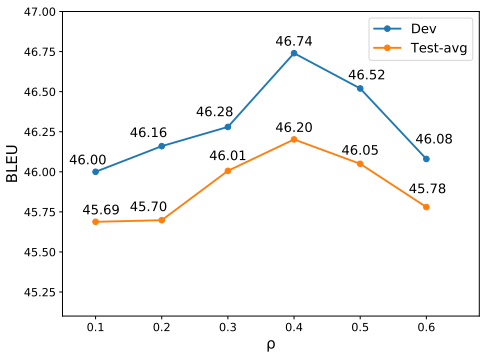

Figure 3: The effect of hyper-parameter ρ on NIST Zh-En translation task.

Strategies	BPE target			Dev	Test
	Dict	Src	Tgt		*Avg*
None	✗	✗	✗	43.94	43.68
Standard	✗	✓	✓	45.16	44.75
Dict	✓	✓	✓	45.71	44.84
Selective	✗	✓	***S***	46.74	46.20

Table 2: The BLEU scores of different BPE strategies. For a BPE target (Dict means dictionary words, Src means source words, and Tgt means target words). ✓, ✗ and ***S*** denote applying BPE, not applying BPE, and applying selective BPE, respectively.

more challenging to identify the proper candidate to be copied.

Finally, PDC substantially outperforms Single-Copy and Flat-Copy, with improvements of 1.66 and 2.20 average BLEU points, due to our effective hierarchical copy mechanism that connects the *Pointer* and the *Disambiguator*, which will be further analyzed in the next sections.

5.2 Effectiveness of Pointer

What distinguishes our *Pointer* from its counterparts of other NMT models is the utilization of semantic information of translation candidates in dictionaries. To verify the effectiveness of this technical design, we implement a PDC variant named **PDC(w/o Dict-Pointer)** whose *Pointer* locates source words based on the encoder state ($\mathbf{h}$) of the vanilla Transformer instead of the dictionary-enhanced encoder state ($\mathbf{h}'$). So the semantic information from dictionaries is not incorporated into the pointing step.

As expected, the performance of PDC(w/o Dict-Pointer) demonstrates a decrement of nearly 1.0 average BLEU score on the test sets compared with PDC, verifying the promising effect of *Pointer*. The results also justify our intuition that the rich information of source words' translations in dictionaries helps to point the proper source word.

5.3 Effectiveness of Disambiguator

To investigate the effectiveness of our bi-view *Disambiguator*, we implement another two model variants: **PDC(w/o Src-View)** that is removed source-view disambiguation and **PDC(w/o Tgt-View)** that is removed target-view disambiguation. As Table 1 shows, the performance of both models significantly decrease.

To further investigate the collaboration between the source-view and target-view disambiguation, we analyze the impact of the hyper-parameter ρ, which denotes how to weight the disambiguation distribution generated from source-view and target-view. In Figure 3, the orange polyline shows the BLEU scores on the development set (MT06), and the blue polyline shows average BLEU scores on another five test sets. By looking into these two polylines' trends, we find that PDC is best-performed when ρ is 0.4, indicating neither the source view nor the target view can be ignored or overly dependent.

These findings prove that both views' contextual information is critical and complementary to identify a specific source word's proper translation, and our *Disambiguator* synthesizes them effectively.

5.4 Effectiveness of Selective BPE

We demonstrate the effects of different BPE strategies in Table 2, where ***None*** does not use BPE at all, ***Standard*** adopts the same BPE strategy as dictionary-independent NMT models, ***Dict*** simply apply BPE to dictionary candidates in addition to standard BPE, and ***Selective*** is our Selective BPE. More detailed settings of each strategy can be found in Table 2, from which we can also clearly observe the superiority of our selective BPE strategy. We attribute this superiority to the fine-grained collaboration between selective BPE and dictionaries, which implicitly yet effectively leveraging the information of which dictionary candidate are likely to be copied.

It is worth mentioning that selective BPE on the target side will not prevent overcoming morphological variance compared with standard BPE. A morphologically inflected target word can be generated in two ways in our system. Firstly, if the target word is not in the candidate set, we will perform standard BPE decomposition. In this scenario, se-

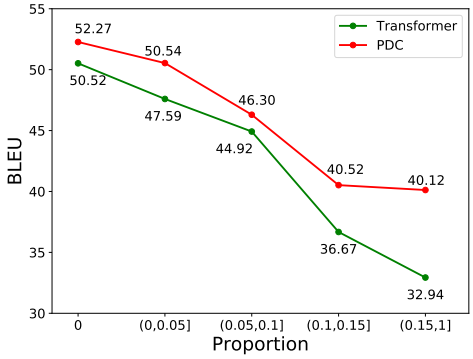

Figure 4: Performance of Transformer and PDC on each subset with different rare word proportions. The figure is plotted based on the MT02 test set results.

lective BPE is the same as standard BPE, and the target word will be generated in a standard way. Otherwise, if the target word is in the candidate set, it will not be decomposed and our method will encourage the model to copy this word directly. Thus, the morphological variance problem can be simply solved by copying.

5.5 Alleviation of the Rare Words Problem

We notice that most dictionary-based NMT works aim to address the rare words problem. Though our work focuses on improving the overall process of incorporating dictionary information as external knowledge, we also conduct a rough experiment to see how our method alleviates the rare words problem.

Specifically, we treat a source word as a rare word if it appears less than ten times in the training set. Then we split the test set into subsets according to the rare word proportions of source sentences. The performance on the subsets is shown in Figure 4. We find that PDC outperforms Transformer by a larger gap on the test subsets with more rare words (e.g., 7.18 for the proportion greater than 0.15), demonstrating that PDC can well alleviate the rare words issue. This observation is also consistent with previous investigations (Luong et al., 2015).

5.6 Results on IWSLT and KFTT

To verify PDC's generalization capability, we further conduct experiments on the IWSLT Zh-En translation task and KFTT En-Ja translation task. Due to space limitations, here we only report the performance of PDC and Transformer. PDC's superiority can be easily observed from the results in Table 3, indicating that PDC can be effectively applied in translation tasks of different language pairs and domains (e.g., news, speech and Wiki).

Method	IWSLT	KFTT
Transformer	19.26	30.12
PDC	20.71	32.18

Table 3: Results on the tasks of IWSLT Zh-En translation and KFTT En-Ja translation.

6 Related Work

6.1 Dictionary-enhanced NMT

Due to the rich prior information of parallel word pairs in bilingual dictionaries, many researchers have dedicated efforts to incorporating bilingual dictionaries into NMT systems. They either generate pseudo parallel sentence pairs based on bilingual dictionaries to boost training (Zhang and Zong, 2016; Zhao et al., 2020), or exploit the bilingual dictionaries as external resources fed into neural networks (Luong et al., 2015; Gulcehre et al., 2016; Arthur et al., 2016; Zhang et al., 2017b; Zhao et al., 2018a,b, 2019b). Our work can be categorized into the second direction, and focus on improving the overall process of incorporating bilingual dictionaries as external knowledge into the latest NMT systems.

In particular, Luong et al. (2015); Gulcehre et al. (2016) first employed copy mechanism (Gu et al., 2016) into NMT to address rare words problem with one-to-one external bilingual dictionaries. Arthur et al. (2016); Zhao et al. (2018a) exploited the prior probabilities from external resource to adjust the distribution over the decoding vocabulary. (Zhao et al., 2018b, 2019b) leverage statistics-based pre-processing method to filter out troublesome words and perform disambiguation on multiple candidates. Our work extends the above ideas and reforms the overall process into a novel end-to-end framework consisting of three steps: pointing, disambiguating, and copying.

6.2 CopyNet

CopyNet is also widely used in text summarization (See et al., 2017; Zhu et al., 2020), automatic post-editing (Huang et al., 2019), grammar correction (Zhao et al., 2019a) and so on.

From a high-level perspective, our methods share a similar Transformer-based architecture with Huang et al. (2019) and Zhu et al. (2020). Huang et al. (2019) employed CopyNet to copy from a draft generated by a pre-trained NMT system. Zhu

et al. (2020) proposed a method that integrates the operation of attending, translating, and summarizing to do cross-lingual summarization. What distinguishes our PDC from other copy-based architectures lies in that the three novel components (***P**ointer*, ***D**isambiguator* and ***C**opier*) and the selective BPE strategy can make full and effective use of dictionary knowledge.

7 Conclusion

We have presented PDC, a new method to incorporate bilingual dictionaries into NMT models, mainly involving four techniques. (1) By integrating semantic information of dictionaries, the enhanced context representations help to locate source words whose dictionary translations will potentially be used. (2) The source and target information is well synthesized and contribute to identifying the optimal translation of a source word among multiple dictionary candidates, in a complementary way. (3) The above two steps are then systematically integrated based on a hierarchical copy mechanism. (4) We finally equip the architecture with a novel selective BPE strategy carefully-designed for dictionary-enhanced NMT.

Experiments show that we achieve competitive results on the Chinese-English and English-Japanese translation tasks, verifying that our approach favorably incorporates prior knowledge of bilingual dictionaries.

Acknowledgements

We thank anonymous reviewers for valuable comments. This research was supported by the National Key Research And Development Program of China under Grant No.2019YFB1405802 and the central government guided local science and technology development fund projects (science and technology innovation base projects) under Grant No.206Z0302G.

References

Philip Arthur, Graham Neubig, and Satoshi Nakamura. 2016. Incorporating discrete translation lexicons into neural machine translation. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1557–1567.

Lei Jimmy Ba, Jamie Ryan Kiros, and Geoffrey E. Hinton. 2016. Layer normalization. *CoRR*, abs/1607.06450.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. In *3rd International Conference on Learning Representations, ICLR 2015*.

Mauro Cettolo, Christian Girardi, and Marcello Federico. 2012. Wit3: Web inventory of transcribed and translated talks. In *Proceedings of the 16th Annual conference of the European Association for Machine Translation*, pages 261–268.

Yong Cheng, Lu Jiang, and Wolfgang Macherey. 2019. Robust neural machine translation with doubly adversarial inputs. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4324–4333.

Yo Joong Choe, Kyubyong Park, and Dongwoo Kim. 2020. word2word: A collection of bilingual lexicons for 3,564 language pairs. In *Proceedings of The 12th Language Resources and Evaluation Conference*, pages 3036–3045.

Jiatao Gu, Zhengdong Lu, Hang Li, and Victor OK Li. 2016. 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)*, pages 1631–1640.

Caglar Gulcehre, Sungjin Ahn, Ramesh Nallapati, Bowen Zhou, and Yoshua Bengio. 2016. Pointing the unknown words. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 140–149.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778.

Xuancheng Huang, Yang Liu, Huanbo Luan, Jingfang Xu, and Maosong Sun. 2019. Learning to copy for automatic post-editing. 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)*, pages 6124–6134.

Minh-Thang Luong, Ilya Sutskever, Quoc Le, Oriol Vinyals, and Wojciech Zaremba. 2015. Addressing the rare word problem in neural machine translation. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 11–19.

Abigail See, Peter J Liu, and Christopher D Manning. 2017. Get to the point: Summarization with pointer-generator networks. In *Proceedings of the 55th Annual Meeting of the Association for Computational*

Linguistics (Volume 1: Long Papers), pages 1073–1083.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. 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)*, pages 1715–1725.

Ilya Sutskever, Oriol Vinyals, and Quoc V Le. 2014. Sequence to sequence learning with neural networks. *Advances in neural information processing systems*, 27:3104–3112.

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

Jianhao Yan, Fandong Meng, and Jie Zhou. 2020. Multi-unit transformers for neural machine translation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1047–1059.

Jian Yang, Shuming Ma, Dongdong Zhang, Zhoujun Li, and Ming Zhou. 2020. Improving neural machine translation with soft template prediction. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5979–5989.

Jiacheng Zhang, Yanzhuo Ding, Shiqi Shen, Yong Cheng, Maosong Sun, Huanbo Luan, and Yang Liu. 2017a. Thumt: An open source toolkit for neural machine translation. *arXiv preprint arXiv:1706.06415*.

Jiacheng Zhang, Yang Liu, Huanbo Luan, Jingfang Xu, and Maosong Sun. 2017b. Prior knowledge integration for neural machine translation using posterior regularization. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1514–1523.

Jiajun Zhang and Chengqing Zong. 2016. Bridging neural machine translation and bilingual dictionaries. *CoRR*, abs/1610.07272.

Wei Zhao, Liang Wang, Kewei Shen, Ruoyu Jia, and Jingming Liu. 2019a. Improving grammatical error correction via pre-training a copy-augmented architecture with unlabeled data. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 156–165.

Yang Zhao, Yining Wang, Jiajun Zhang, and Chengqing Zong. 2018a. Phrase table as recommendation memory for neural machine translation. In *Proceedings of the 27th International Joint Conference on Artificial Intelligence*, pages 4609–4615.

Yang Zhao, Jiajun Zhang, Zhongjun He, Chengqing Zong, and Hua Wu. 2018b. Addressing troublesome words in neural machine translation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 391–400.

Yang Zhao, Jiajun Zhang, Yu Zhou, and Chengqing Zong. 2020. Knowledge graphs enhanced neural machine translation. In *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI 2020*, pages 4039–4045. ijcai.org.

Yang Zhao, Jiajun Zhang, Chengqing Zong, Zhongjun He, and Hua Wu. 2019b. Addressing the under-translation problem from the entropy perspective. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 451–458.

Junnan Zhu, Yu Zhou, Jiajun Zhang, and Chengqing Zong. 2020. Attend, translate and summarize: An efficient method for neural cross-lingual summarization. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1309–1321.