Automatic Paraphrasing via Sentence Reconstruction and Back-translation

Abstract

Paraphrase generation plays key roles in NLP tasks such as question answering, machine translation, and information retrieval. In this paper, we propose a novel framework for paraphrase generation. It simultaneously decodes the output sentence using a pretrained wordset-to-sequence model and a back-translation model. We evaluate this framework on Quora, WikiAnswers, MSCOCO and Twitter, and show its advantage over previous state-of-the-art unsupervised methods and distantly-supervised methods by significant margins on all datasets. For Quora and WikiAnswers, our framework even performs better than some strongly supervised methods with domain adaptation. Further, we show that the generated paraphrases can be used to augment the training data for machine translation to achieve substantial improvements.

1 Introduction

The paraphrase of a sentence retains its meaning but makes different choices of words and expressions than the original form. Paraphrase generation plays an important role in many downstream tasks, such as question answering, machine translation, and information retrieval [Hu *et al.*, 2019a].

Most existing parallel datasets for paraphrase generation are domain-specific. Quora and WikiAnswers [Fader *et al.*, 2013] datasets only contain questions; sentences in MSCOCO [Lin *et al.*, 2014] dataset are mostly descriptions for physical objects since they are the captions of images; and PPDB [Ganitkevitch *et al.*, 2013] contains phrases rather than sentences. The performance of a model trained with these domain-specific parallel data declines seriously when it is used in another domain [Li *et al.*, 2019].

Many efforts were made to solve this domain adaptation problem. These efforts are roughly divided into three directions: unsupervised fine-tuning for supervised model, unsupervised methods based on word/phrase replacement, and distantly-supervised methods based on bilingual data. Li *et al.* [2019] chose to fine-tune the supervised model with non-parallel in-domain data, but the performance of their model decreases a lot when the domain span is large. Liu *et al.* [2019] and Miao *et al.* [2019] used unsupervised methods to generate paraphrases, but their models are mostly based on the variation of words and phrases and can hardly change the structure of the whole sentence. Wieting and Gimpel [2017] generated paraphrases with a back-translation model, but the existing translation models are sometimes not very accurate, which also affects the performance of their method. Liu *et al.* [2020] use bilingual data to train an unsupervised model, but their improvement is mainly brought about by the follow-up supervised fine-tuning.

In this paper, we propose a novel paraphrase generation framework that does not require any parallel paraphrase data and can be applied in any domain. In our framework, two kinds of underlying semantics are extracted from the original sentence and are recombined into a new sentence through a hybrid decoder.

The first kind of underlying semantics is represented by a word set, which is inspired by the Denoising Auto-Encoder (DAE) [Vincent *et al.*, 2008]. A bag of words is a great carrier of information, as it communicates the central idea without syntactic constraints. People can produce different sentences with similar meaning from the same set of words. Table 1 shows an example of such paraphrase sentences. We construct a word set from the original sentence and extend the word set into a complete sentence with a set-to-sequence (set2seq) model, which is adapted from the well-known sequence-to-sequence (seq2seq) model by ignoring the sequential information from the input sequence.

word set: **(man, sit, bike, bench)**
A *man* is *sitting* on a *bench* next to a *bike*
A *man* is *sitting* on a *bench* next to a *bicycle*
A *man sits* on a *bench* by a *bike*
Man sitting on a *bench* near a personal *bicycle*
A *man* is *sitting* on a *bench* with a *bike*

Table 1: An example of paraphrases formed from the same set of words (enclosed in the parentheses).

The second carrier of semantics is the translation of the original sentence into another language. Semantics is preserved but syntactic perturbations are added when the translation is then translated back to the original language. This is known as back-translation [Wieting and Gimpel, 2017]. [1]

[1]Considering that training the back-translation model requires

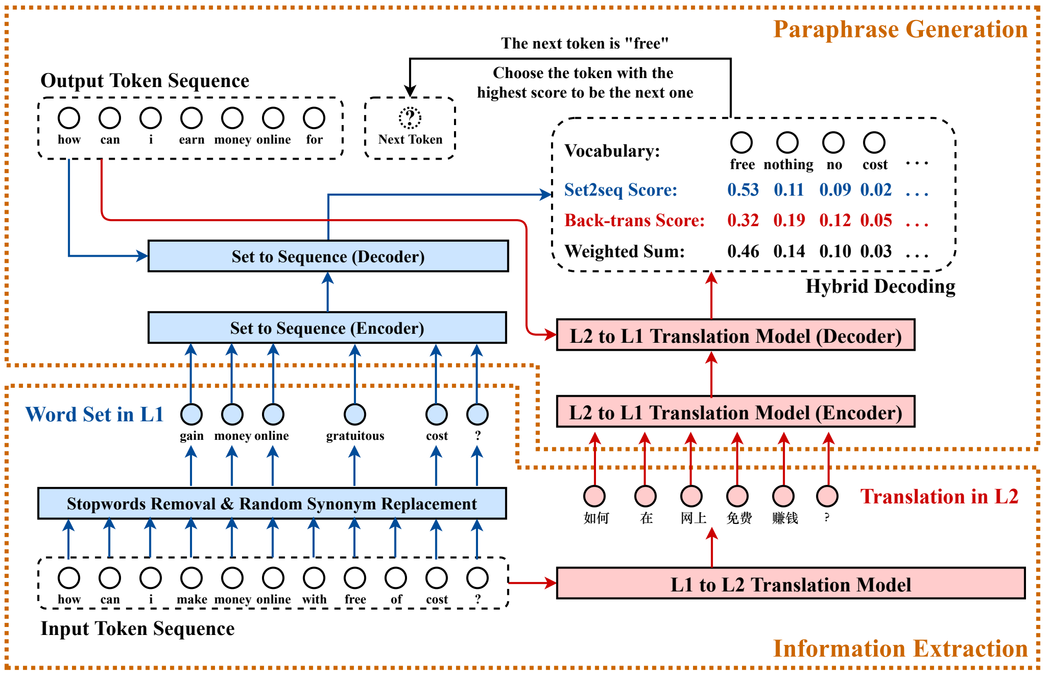

Figure 1: Our Paraphrasing Framework

The above two types of semantics are complementary. The back-translation makes up for the missing information in the set2seq model, such as sequential information. The set2seq model gives the back-translation model some lexical hints and makes the translation result more accurate. We thus integrate the decoding parts of the set2seq model and the back-translation model to jointly generate paraphrases.

We evaluate our framework on four paraphrasing datasets, namely Quora, WikiAnswers, MSCOCO, and Twitter [Lan *et al.*, 2017], and achieve the state-of-the-art accuracies compared to existing models trained with non-parallel data.

We also train the set2seq model on a big common-domain dataset and test it on these four datasets, and still obtain decent results. We call the set2seq model trained from the big common-domain dataset "set2seq-common", and can apply it to any domain when there is no in-domain data to train a set2seq model.

Finally, we propose an application of our paraphrase generator: to augment the training data of a neural machine translation (NMT) model between low-resource languages and English. We paraphrase the English sentences in the parallel training pairs with set2seq-common and improve the BLEU score of X-to-English translation by 1.53 to 2.17, where X is a low-resource language.

In summary, the main contributions of this work are:

- We are the first to apply the set2seq model to the task of paraphrase generation by combining it with a back-translation model through a hybrid decoder.
- The framework proposed by us achieve state-of-the-art accuracies on four benchmark datasets compared with existing methods.
- We apply our method to augment the training data of low-resource translation tasks and obtain significant improvement in translation quality.

intensive computing resources, we will open-source these models and the codes when the paper is published.

2 Approach

In this section, we describe our framework. We first give an overview and then describe the detailed components of the framework.

2.1 Overview

The set2seq model and the two translation models used in back-translation are trained separately, and our framework is designed for use during inference time only. Figure 1 shows the architecture for our framework, which is divided into two major components and two major phases. The two components are sentence reconstruction based on word set, and back-translation. The two phases are information extraction and paraphrase generation.

Suppose the original sentence is in language L_1 and the back-translation is via language L_2. During information extraction phase, given an input sequence of tokens $X = [x_1, x_2, \cdots]$, we process it in two different approaches to extract two different representations of the underlying semantics: a word set and a translation in language L_2. For the former, we construct a word set $WS = \{w_1, w_2, \cdots\}$. For the latter, we use a L_1-L_2 translation model to get a sequence of translated tokens $Z = [z_1, z_2, \cdots]$ in L_2.

In the paraphrase generation phase, we employ a hybrid decoder which takes inputs from two separate encoders, one from the set2seq model and the other from the L_2-L_1 translation model. We encode the word set WS and the L_2 token sequence Z respectively to obtain two hidden states H_{ws} and H_{bt}. The hybrid decoder maintains a single output sequence, generating one token at each step based on H_{ws}, H_{bt}, and the previously generated tokens.

2.2 Word Set Constructor

We use the word set constructor to extract a word set from the original sentence. To ensure accuracy and diversity of sentences generated from the word set, the word set constructor tries to strike a balance between both content preservation and lexical variation.

For content preservation, we could select informative words from the original sentence by either removing stopwords or retaining high-IDF words to build the keywords set KWS, which will be passed to the next stage. Here, we choose to remove stopwords, the reason for which will be explained in the result analysis in Section 3.5

To increase the lexical diversity of the generated paraphrase, each word in KWS is randomly replaced with one of its synonyms using WordNet [Miller, 1995], and optionally, itself. This process is known as "random replacement". We obtain WS after this step. BERT based methods, instead of WordNet, can also be used to generate synonyms. They are not used because: i) we have to generate synonyms for every single word in the training set, and it is too computational expensive if we use BERT; and ii) WordNet is good enough for generating high-quality word sets.

2.3 Set-to-Sequence

A set2seq model consists of an encoder and a decoder, similar to a seq2seq model. However, instead of taking a sequence as the input, the input of a set2seq model is a set of tokens with no sequential information.

To train a set2seq model, we prevent the encoder to do serial processing for the input set. RNN-based models are inappropriate for this purpose due to their recurrent nature. Therefore we use a transformer-based model. In transformer, the sequential information of the input sequence is captured in the position encoding. We use a transformer but omit the position encoding in the encoder as the set2seq model.

We train set2seq with word set WS as the input and original sentence X as the output. This training data is automatically created and thus the training process is considered self-supervised. Specifically, given a set of words $WS = \{w_1, w_2, \cdots\}$, the set2seq model does the following steps in a single layer while encoding:

$$\overline{h_i} = \text{LayerNorm}(\text{MultiAttn}(h_i)) + h_i \quad (1)$$

$$h_{i+1} = \text{LayerNorm}(\text{FF}(h_m)) + \overline{h_i}, \quad (2)$$

where h_{i+1} is the output of layer i and h_0 is the embedding of tokens in WS.

2.4 Hybrid Decoding

A hybrid decoder can take the hidden states of multiple encoders as input and generate a single output sequence based on the information from all hidden states.

As we mentioned before, we divide the framework into two components, the set2seq model and the back-translation model, and obtain two hidden states H_{ws} and H_{bt}.

Assume that our vocabulary is $V = \{v_1, v_2, \cdots, v_D\}$ with D different tokens. In decoding step t, the decoder of the set2seq and the L_2-L_1 translation model can give the probability of v being the next token individually. Supposing we already generated $t-1$ tokens $y_1, y_2, ...y_{t-1}$, the next token y_t to be generated is given by the following equation:

$$y_t = \arg\max_{v \in V} \big(P_{bt}\left(v_i | y_{1:t-1}, H_{bt}\right) + \lambda \cdot P_{ws}\left(v_i | y_{1:t-1}, H_{ws}\right) \big) \quad (3)$$

Here P_{ws} and P_{bt} are the probabilities of v_i being the next token calculated by the decoder of the set2seq model and the L_2-L_1 translation model respectively, and λ is the hyperparameter to balance the weight between the two probabilities.

3 Experimental Results

In this section, we first introduce the experimental setup, including datasets, baselines, evaluation metrics, and implementation details. Then, we show the results the competing methods. Finally, we analyze the results from different aspects.

3.1 Datasets

We evaluate our framework on four different datasets, namely Quora, WikiAnswers, MSCOCO, and Twitter. Following Liu *et al.* [2019], we randomly choose 20K parallel paraphrase pairs as the test set and 3K parallel paraphrase pairs as the validation set for Quora, WikiAnswers, and MSCOCO.

Training with In-domain Data. We randomly sample the remaining parallel paraphrase pairs and pick one sentence from each pair to construct the non-parallel training data. The number of selected sentences is the same as the work by Liu *et al.* [2019], which is 400K for Quora[2], 500K for WikiAnswers, 320K for MSCOCO and 110K for Twitter.

Training with Common-Domain Data. When there is no sufficient available target-domain non-parallel data , it is hard to train unsupervised models or fine-tune supervised models in the target-domain. Our solution is to train the set2seq model with a big common-domain dataset and apply it to the target-domain. We name the model "set2seq-common". We test the performance of our framework with set2seq-common on four datasets to show the generality of our framework. Further, we apply set2seq-common in the Application section.

3.2 Baselines and Evaluation Metrics

We compare our framework with five unsupervised/distantly-supervised methods and four supervised methods with domain adaptation. We re-produce ParaNMT [Wieting and Gimpel, 2017] and ParaBank [Hu *et al.*, 2019b] using our translation models, and take the results from Liu *et al.* [2019] and Liu *et al.* [2020] for other baselines. For a fair comparison, we keep the scripts for data pre-processing and evaluation from UPSA.

[2] https://data.quora.com/First-Quora-Dataset-Release-Question-Pairs

On the Quora dataset, we even use the same train-test split as UPSA.[3]

Unsupervised and distantly-supervised methods. The current state-of-the-art unsupervised method is Unsupervised Paraphrasing by Simulated Annealing (UPSA) [Liu *et al.*, 2019], which is also our main target of comparison. The other unsupervised methods is CGMH [Miao *et al.*, 2019]. Distantly-supervised baselines are the unsupervised part by Liu et al. [Liu *et al.*, 2020], ParaNMT and ParaBank(-3rd IDF). Note that ParaNMT used back-translation to generate paraphrases, so it can be viewed as "back-translation only".

Supervised methods with domain adaptation. Decomposable Neural Paraphrase Generation (DNPG) [Li *et al.*, 2019] is the current state-of-the-art method for supervised paraphrase generation. Other baselines are Pointer-generator [See *et al.*, 2017], Transformer [Vaswani *et al.*, 2017] with copy mechanism, and MTL[Domhan and Hieber, 2017] with copy mechanism.

Evaluation metrics. For fair comparisons, we take the same evaluation metrics as in UPSA and DNPG, which are iBLEU [Sun and Zhou, 2012], BLEU [Papineni *et al.*, 2002] and ROUGE [Lin, 2004] scores. BLEU and ROUGE scores are common evaluation metrics for NLG tasks while iBLEU is especially designed for paraphrase generation tasks. It penalizes the similarity between paraphrase and the original sentence. Suppose the input sentence is src, the output paraphrase is out, and the ground truth paraphrase is trg, we calculate iBLEU as follows:

$$\text{iBLEU} = \alpha \cdot \text{BLEU}(out, trg) - (1-\alpha) \cdot \text{BLEU}(out, src) \quad (4)$$

BLEU and ROUGE only consider the accuracy but ignore the diversity of generated paraphrases, while iBLEU considers both. So we use iBLEU as our main evaluation metric. We set $\alpha = 0.9$, same as other baselines.

3.3 Implementation and Training Details

To be consistent with the pre-processing of UPSA and DNPG, we convert the input words into lower-case and truncate all sentences to up to 20 words. For the convenience of hybrid decoding, we learn a shared byte-pair encoding (BPE, [Sennrich *et al.*, 2016]) with size 50k from the training data for translation models, and use a 30K vocabulary for all models. The hyper-parameter λ in the hybrid decoder is set to 0.5 for all datasets after experimenting with difference choices.

For the translation models in back-translation, we train them with the WMT17[4] zh-en dataset [Ziemski *et al.*, 2016] with a standard transformer for 3 days on two GTX-2080 GPUs. We reuse these translation models for ParaNMT and ParaBank. For the set2seq-common model, we use the news-crawl-2016 English monolingual data from WMT17 and train 1.5 days with a standard transformer. For the domain-specific set2seq models, we use a 2-layer transformer with 300 embedding size, 256 units, 1024 feed-forward dimensions for all layers to train them. The training lasts 3 hours on a single GTX-2080 GPU. Set2seq is a lightweight model with 31M parameters, 3.7M parameters for multi-head attention layers, only one-third of a standard transformer.

To calculate iBLEU and BLEU, four references are used for MSCOCO, five for WikiAnswers, and one for other datasets. Some test cases in WikiAnswers may have fewer than 5 references. For ROUGE scores, we take the average score against all references.

3.4 Results

Table 2 presents our experimental results. We compare three different models with the previous methods, namely set2seq, set2seq-common+BT, and set2seq+BT, where BT stands for back-translation. We show the set2seq alone here to demonstrate that the useful information comes not only from the translation, since the set2seq model alone can already outperform almost all competitors. Our framework outperforms all existing unsupervised methods, distantly-supervised methods, and supervised methods with domain adaptation.

3.5 Analysis

Datasets. Due to the domain-specific differences between four datasets, it is understandable that scores on all metrics vary a lot across different datasets.

Paraphrases from MSCOCO are descriptions of images, the set2seq model fits this dataset quite well since the process of generating paraphrases are similar: one extends information from a static picture; the other extends from a word set.

Lack of training data for Twitter leads to insufficient training of most models. Models using back-translation perform extraordinary well since they have adequate information. Besides, set2seq-common+BT achieves an excellent result, which shows the advantages of the set2seq-common model compared with the set2seq model trained with insufficient in-domain data.

Ablation Study. Table 3 shows the result of the ablation study on the Quora dataset, where BLEU$_{ref}$ is the BLEU between reference and output, the higher the better and BLEU$_{src}$ is the BLEU between source sentence and output, the lower the better.

We demonstrate that removing stopwords is better than keeping high-IDF words. For high-IDF words, we keep the top $k\%$ high-IDF words in the original sentence. We set $k = 50$, the best from $\{30, 40, 50, 60, 70\}$ by empirics. We also tried TextRank [Mihalcea and Tarau, 2004] to score words and get similar results with IDF scores.

Removing random replacement and adding position encoding can both give high BLEUs between reference sentences and output paraphrases, but substantially reduce the diversity of the generated sentences.

Human Evaluation. We choose 100 sentences from Quora and ask 3 human annotators to score the results from different methods blindly on a scale of 1 to 5 according to fluency and accuracy (the higher the better). Fluency measures whether the paraphrase conforms to grammar and common sense; accuracy measures whether the paraphrase has the same meaning as the original sentence though in a different expression.

We can see that word/phrase based methods have bad performances on fluency since their language model is trained

[3] https://github.com/anonymity-person/UPSA

[4] http://statmt.org/wmt17/translation-task.html

	Model	Quora iBLEU	Quora BLEU	Quora R-1	Quora R-2	WikiAnswers iBLEU	WikiAnswers BLEU	WikiAnswers R-1	WikiAnswers R-2
Supervised	DNPG (SOTA)	18.01	25.03	63.73	37.75	34.15	41.64	57.32	25.88
Supervised + Domain-Adapted	Pointer-generator	5.04	6.96	41.89	12.77	21.87	27.94	53.99	20.85
	Transformer+Copy	6.17	8.15	44.89	14.79	23.25	29.22	53.33	21.02
	MTL+Copy	7.22	9.83	47.08	19.03	21.87	30.78	54.10	21.08
	DNPG	10.39	16.98	56.01	28.61	25.60	35.12	56.17	23.65
Unsupervised	CGMH	9.94	15.73	48.73	26.12	20.05	26.45	43.31	16.53
	UPSA	12.02	18.18	56.51	30.69	24.84	32.39	54.12	21.45
Distantly-Supervised	Liu *et al.* [2020]	9.90	15.03	52.65	23.18	-	-	-	-
	ParaNMT(back-translation)	10.69	15.75	52.28	25.12	14.94	20.01	30.55	10.23
	ParaBank	9.92	14.71	50.03	23.80	13.14	17.56	28.97	9.34
	set2seq (ours)	13.54	20.85	58.27	32.59	25.98	33.41	55.95	23.08
	set2seq-common+BT (ours)	12.60	18.85	57.13	31.19	25.04	33.43	55.81	23.12
	set2seq+BT (ours)	**14.66**	**22.53**	**59.98**	**34.09**	**28.27**	**37.42**	**56.71**	**24.94**

	Model	MSCOCO iBLEU	MSCOCO BLEU	MSCOCO Rouge1	MSCOCO Rouge2	Twitter iBLEU	Twitter BLEU	Twitter Rouge1	Twitter Rouge2
Unsupervised	CGMH	7.84	11.45	32.19	8.67	4.18	5.32	19.96	5.44
	UPSA	9.26	14.16	37.18	11.21	4.93	6.87	28.34	8.53
Distantly-Supervised	Liu *et al.* [2020]	6.67	9.86	22.14	6.21	-	-	-	-
	ParaNMT(back-translation)	7.39	10.71	30.74	8.68	7.57	10.79	35.38	14.74
	ParaBank	6.45	9.48	29.22	8.35	6.50	9.71	34.56	13.92
	set2seq (ours)	**11.54**	17.61	39.87	13.67	5.72	7.48	31.65	10.89
	set2seq-common+BT (ours)	9.07	13.44	35.90	11.05	9.73	**14.30**	**39.23**	**18.82**
	set2seq+BT (ours)	11.39	**17.93**	**40.28**	**14.04**	**9.95**	13.97	38.96	18.32

Table 2: Evaluation results on Quora, WikiAnswers, MSCOCO and Twitter. The comparison with supervised + domain adapted methods is only on Quora and WikiAnswers because results of current SOTA method (DNPG) are only available on these two datasets. The previous highest scores are marked with the underlines and the present highest scores are marked with the bold font. The supervised method DNPG (SOTA) is shown here only for reference.

Model Variants	iBLEU	BLEU$_{ref}$	BLEU$_{src}$
set2seq+BT	**14.66**	22.53	**56.17**
$\ominus$ excluding stopwords $\oplus$ retaining high-IDF	13.46	22.15	64.75
$\ominus$ random replacement	13.78	**23.92**	77.47
$\oplus$ position encoding	14.07	23.26	68.60

Table 3: Ablation Study on Quora.

Method	Accuracy Score	Accuracy Agreement	Fluency Score	Fluency Agreement
CGMH	3.15	0.55	3.42	0.50
UPSA	3.49	0.54	3.51	0.55
DNPG(Adapted)	3.32	0.48	3.62	0.54
Back Translation	3.37	0.59	**4.18**	0.58
set2seq+BT(ours)	**3.78**	0.57	4.13	0.55

Table 4: Results for Human Evaluation (Mean and Kappa).

on a small dataset. Paraphrases generated by back-translation are not very accurate since they are not trained by in-domain data. By both fluency and accuracy, our method performs the overall best.

4 Application on Translation Tasks

We apply our paraphrase generator to augment the training data of X-English translation task, where X is a low-resource language. Since it is difficult to find high-quality test sets for low-resource languages, we use three commonly-studied languages and reduce their parallel training pairs to 150k and 300k to simulate low-resource languages.

4.1 Data Augmentation

The size of training pairs for NMT tasks is of great importance to the final result. However, it is quite expensive to enlarge the dataset manually due to the high labor price and the huge workload, so data augmentation for NMT is a popular research topic in recent years. To this end, in the task of translating low-resource languages to English, we paraphrase the sentence in English to augment training pairs automatically.

For each language, we carry on two experiments with 150k data and 300k data respectively. For each experiment, we train the model with original data as the baseline. For each experiment, we train the model with the original data as the

	Size	Orig. Pairs	Augmented
De-En	150k	12.89	15.06
	300k	15.67	17.20
Zh-En	150k	10.21	11.99
	300k	12.10	14.07
Ru-En	150k	16.88	18.55
	300k	19.30	21.09

Table 5: BLEU scores of translating three languages into English; each task is trained with 150k/300k original pairs and 3M/6M pairs after data-augmentation.

baseline.

Regarding augmentation, we make 10 copies of the original sentences, construct 10 word sets with different seeds in random replacement from the 10 copies and generate 10 paraphrases with set2seq-common+BT. To increase the diversity of the results, we use random sampling [Edunov *et al.*, 2018] during decoding. We take the 10 copies and 10 paraphrases as the augmented data.

For the set2seq-common model, since the sentences in the NMT training set is longer, we truncate all sentences to 50 words instead of 20 during the training stage and do not truncate any sentences during the inference stage.

4.2 Experimental Setup and Results

We experiment on German-English (de-en), Chinese-English (zh-en), and Russian-English (ru-en) translation pairs. For the training data, we obtain the de-en data from WMT17-europarl[5][Koehn, 2005], and the ru-en data from WMT17 news-commentary and zh-en data from LDC [Liberman, 2002; Huang *et al.*, 2002]. The reason for not using zh-en data from WMT17 is that we are already using the zh-en pairs from WMT17 to train the translation models. For test sets, there are 3004 pairs for de-en, 2000 pairs for zh-en and 3000 pairs for ru-en from the WMT17 news-test.

For each language, we learn a shared BPE of size 50,000 and extract vocabulary of up to 50,000 from the training set for both English and the target language with the shared BPE.

We train translation models with a standard transformer-base model [Vaswani *et al.*, 2017]. For the result of each model, we take the average of test results from 5 checkpoints after convergence.

Table 5 shows the result. Paraphrase augmentation improves the model trained with original data pairs by anywhere from 1.53 to 2.17 BLEU.

When producing paraphrases, our methods do use additional data, such as monolingual English data and Chinese English translation data. It is conceivable that there exists other advanced NMT methods that use these data in different ways. However, the purpose of this section is to show the effectiveness of our long-sentence paraphrase generation methods, since only accurate and diverse paraphrases can be used as good translation pairs and subsequently train good translators.

[5]http://www.statmt.org/europarl/

5 Related Work

We show the relevant work of paraphrase generation from the aspects of supervised, distance-supervised, and unsupervised methods.

For supervised methods, Prakash *et al.* [2016] proposed "stacked residual LSTM" as the earliest deep-learning method in this topic, seq2seq models like transformer [Vaswani *et al.*, 2017] and MTL [Domhan and Hieber, 2017] outperformed many methods due to the advantages of their model structures. We include these well-known methods in our baseline. Li *et al.* [2019] proposed the current state-of-the-art method DNPG and revealed the disadvantage of supervised methods when it comes to domain adaptation. Other methods include VAE-SVG [Gupta *et al.*, 2018] and transformer-pb [Wang *et al.*, 2019], but these methods perform worse than DNPG and have no discussion about domain adaptation, so we do not include them in our baselines.

For distance-supervised methods, Wieting and Gimpel [2017] created a 50M parallel dataset for paraphrases with back-translation, Hu *et al.* [2019b] used lexically-constrained to improve the diversity of generated paraphrase, and their work is proved to be useful for many downstream tasks like Natural Language Inference [Hu *et al.*, 2019a]. Liu *et al.* [2020] also use bilingual data to generate paraphrase without parallel data. However, their focus is on the supervised fine-tuning part. Their method do not performs well without the fine-tuning part.

For unsupervised methods, Miao *et al.* [2019] used Metropolis-Hastings Sampling to generate paraphrases, Liu *et al.* [2019] generated paraphrases with Simulated Annealing, both of them were the best at their times. We compare our framework with these two methods to show changes on the sentential level are more reliable than changes on the lexical level. Siddique *et al.* [2020] proposed a method for paraphrasing with deep reinforcement learning. However, we do not regard it as a baseline since their results are not convincing enough for the following two reasons:

- By iBLEU in (4), their $BLEU_{src}$ is 28.04 on Quora, and 91.98 on WikiAnswers, which shows a very large and abnormal disparity.
- The authors provided us with their test set on Quora, where the BLEU score between source sentences and references is 69.75. However, the score should be around 25 if the test samples are randomly selected from Quora. With their test set, it is easier to generate paraphrases similar to references.

6 Conclusion

In this paper, we proposed a novel framework for automatical paraphrase generation without parallel training data. It outperforms most existing unsupervised and distantly supervised methods. While the results are positive, some questions remain. Can we find more underlying semantics to represent the input sentence? Can we replace the back-translation model with a lighter-weight model? We plan to look into these questions in the future and generate better paraphrases.

References

[Domhan and Hieber, 2017] Tobias Domhan and Felix Hieber. Using target-side monolingual data for neural machine translation through multi-task learning. In *EMNLP*, 2017.

[Edunov *et al.*, 2018] Sergey Edunov, Myle Ott, Michael Auli, and David Grangier. Understanding back-translation at scale. *arXiv preprint*, 2018.

[Fader *et al.*, 2013] Anthony Fader, Luke Zettlemoyer, and Oren Etzioni. Paraphrase-driven learning for open question answering. In *ACL*, 2013.

[Ganitkevitch *et al.*, 2013] Juri Ganitkevitch, Benjamin Van Durme, and Chris Callison-Burch. Ppdb: The paraphrase database. In *NAACL*, 2013.

[Gupta *et al.*, 2018] Ankush Gupta, Arvind Agarwal, Prawaan Singh, and Piyush Rai. A deep generative framework for paraphrase generation. In *AAAI*, 2018.

[Hu *et al.*, 2019a] J Edward Hu, Huda Khayrallah, Ryan Culkin, Patrick Xia, Tongfei Chen, Matt Post, and Benjamin Van Durme. Improved lexically constrained decoding for translation and monolingual rewriting. In *NAACL*, 2019.

[Hu *et al.*, 2019b] J Edward Hu, Rachel Rudinger, Matt Post, and Benjamin Van Durme. Parabank: Monolingual bitext generation and sentential paraphrasing via lexically-constrained neural machine translation. *arXiv preprint*, 2019.

[Huang *et al.*, 2002] Shudong Huang, David Graff, and George Doddington. *Multiple-translation Chinese corpus*. Linguistic Data Consortium, University of Pennsylvania, 2002.

[Koehn, 2005] Philipp Koehn. Europarl: A parallel corpus for statistical machine translation. In *MT summit*, 2005.

[Lan *et al.*, 2017] Wuwei Lan, Siyu Qiu, Hua He, and Wei Xu. A continuously growing dataset of sentential paraphrases. *arXiv preprint*, 2017.

[Li *et al.*, 2019] Zichao Li, Xin Jiang, Lifeng Shang, and Qun Liu. Decomposable neural paraphrase generation. *arXiv preprint*, 2019.

[Liberman, 2002] Mark Liberman. Emotional prosody speech and transcripts. *http://www. ldc. upenn. edu/Catalog/CatalogEntry. jsp? catalogId= LDC2002S28*, 2002.

[Lin *et al.*, 2014] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *ECCV*, 2014.

[Lin, 2004] Chin-Yew Lin. Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out*, pages 74–81, 2004.

[Liu *et al.*, 2019] Xianggen Liu, Lili Mou, Fandong Meng, Hao Zhou, Jie Zhou, and Sen Song. Unsupervised paraphrasing by simulated annealing. *arXiv preprint*, 2019.

[Liu *et al.*, 2020] Mingtong Liu, Erguang Yang, Deyi Xiong, Yujie Zhang, Chen Sheng, Changjian Hu, Jinan Xu, and Yufeng Chen. Exploring bilingual parallel corpora for syntactically controllable paraphrase generation. IJCAI, 2020.

[Miao *et al.*, 2019] Ning Miao, Hao Zhou, Lili Mou, Rui Yan, and Lei Li. Cgmh: Constrained sentence generation by metropolis-hastings sampling. In *AAAI*, 2019.

[Mihalcea and Tarau, 2004] Rada Mihalcea and Paul Tarau. Textrank: Bringing order into text. In *EMNLP*, 2004.

[Miller, 1995] George A Miller. Wordnet: a lexical database for english. *Communications of the ACM*, 1995.

[Papineni *et al.*, 2002] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. Bleu: a method for automatic evaluation of machine translation. In *ACL*, 2002.

[Prakash *et al.*, 2016] Aaditya Prakash, Sadid A Hasan, Kathy Lee, Vivek Datla, Ashequl Qadir, Joey Liu, and Oladimeji Farri. Neural paraphrase generation with stacked residual lstm networks. *arXiv preprint*, 2016.

[See *et al.*, 2017] Abigail See, Peter J Liu, and Christopher D Manning. Get to the point: Summarization with pointer-generator networks. *arXiv preprint*, 2017.

[Sennrich *et al.*, 2016] Rico Sennrich, Barry Haddow, and Alexandra Birch. Edinburgh neural machine translation systems for wmt 16. *arXiv preprint*, 2016.

[Siddique *et al.*, 2020] A. B. Siddique, Samet Oymak, and Vagelis Hristidis. Unsupervised paraphrasing via deep reinforcement learning. KDD, 2020.

[Sun and Zhou, 2012] Hong Sun and Ming Zhou. Joint learning of a dual smt system for paraphrase generation. In *ACL Short Paper*, 2012.

[Vaswani *et al.*, 2017] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *NIPS*, 2017.

[Vincent *et al.*, 2008] Pascal Vincent, Hugo Larochelle, Yoshua Bengio, and Pierre-Antoine Manzagol. Extracting and composing robust features with denoising autoencoders. ICML, 2008.

[Wang *et al.*, 2019] Su Wang, Rahul Gupta, Nancy Chang, and Jason Baldridge. A task in a suit and a tie: paraphrase generation with semantic augmentation. In *AAAI*, 2019.

[Wieting and Gimpel, 2017] John Wieting and Kevin Gimpel. Paranmt-50m: Pushing the limits of paraphrastic sentence embeddings with millions of machine translations. *arXiv preprint*, 2017.

[Ziemski *et al.*, 2016] Michał Ziemski, Marcin Junczys-Dowmunt, and Bruno Pouliquen. The united nations parallel corpus v1. 0. In *LREC*, 2016.