Unsupervised Paraphrasing via Sentence Reconstruction and Back-translation

Anonymous EMNLP submission

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 unsupervised paraphrase generation. It simultaneously decodes the output sentence through 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 by significant margins on all datasets. For Quora and WikiAnswers, our framework even performs better than some strong 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 gains.

1 Introduction

The paraphrase of a sentence should retain the meaning but change the word expression of the original sentence. 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 objects since they are from captions of images; and PPDB (Ganitkevitch et al., 2013) contains phrases rather than sentences. The performance of a supervised model declines seriously when it comes to another domain (Li et al., 2019). Therefore, unsupervised methods are often used for paraphrase generation.

Existing unsupervised methods are mostly based on the variation of words and phrases and can hardly change the structure of the whole sentence. For example, Liu et al. (2019) proposed a method using simulated annealing for words and phrases, and Miao et al. (2019) used Metropolis Hastings in the word space.

In this paper, we propose a novel unsupervised paraphrase generation framework that can alter the expression at the sentence level. We extract the underlying semantics from the original sentence and extend it into a new sentence. Information loss may occur when extracting semantics. To retain more information of the original sentence, we extract in two different directions and combine the extracted information in a hybrid decoder (Section 2.4) to generate paraphrases.

The first expression of underlying semantics is a word set extracted from the original sentence. Bag of words are great carrier of information, as they harbor the central idea without syntactic constraints. People can generate different sentences of the same meaning from the same word set. 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 (Section 2.3), which is adapted from the well known sequence-to-sequence (seq2seq) model by ignoring the sequential information from the input sequence.

The second carrier of semantics is the translation of the original sentence into another language. Semantics is preserved when the translation is translated back to the original language. This is known as back-translation (Wieting and Gimpel, 2017). We integrate the decoding part of the set2seq model and the back-translation model to 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 unsuper-

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 in red.

vised models.

Domain-adaptation is to train the model with parallel paraphrasing pairs in the source domain and fine-tune the model with non-parallel sentences in the target domain, which can also be considered unsupervised from the perspective of the target domain. Therefore, we also compare our method with domain-adaptation supervised methods with in Quora and WikiAnswers. The comparison is not on all four datasets because results of the SOTA method (Li et al., 2019) are only available on Quora and WikiAnswers.

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", it can be applied to any domain when there is no in-domain data to train an in-domain set2seq model.

We propose an application of our paraphrase generator: to augment the training data of Neural Machine Translation (NMT) 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 our work are:

- We propose a novel framework for unsupervised paraphrasing at the sentence level and achieve state-of-the-art accuracies on four benchmark datasets compared with existing unsupervised methods.
- We show that our framework outperforms most domain-adapted supervised methods including the current state-of-the-art method on two benchmark datasets.
- We apply our method to augment the training data of low-resource translation tasks and obtain significant improvement in translation quality.

2 Approach

In this section, we describe our framework. First, we give an overall description of the framework in Section 2.1. Then, we show how to construct a word set from the original sentence (Section 2.2), how to generate paraphrases from the word set with the set2seq model (Section 2.3), and how to incorporate the set2seq model and the back-translation model (Section 2.4)

2.1 Overview

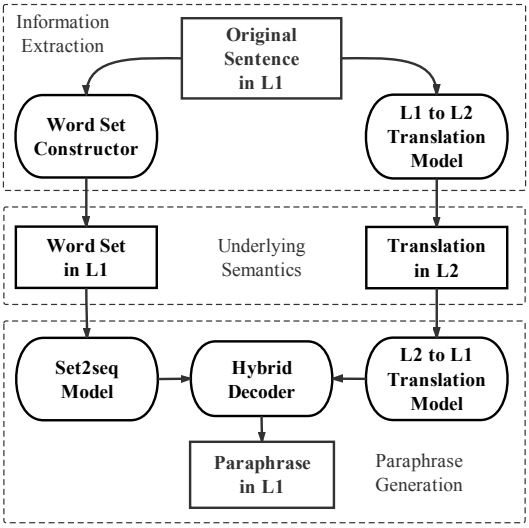

Figure 1: Our Paraphrsing Framework

The set2seq model and the two translation models used in back-translation are trained separately, and our framework is designed for the inference period. Figure 1 shows a macro view of the architecture for our framework, which is divided into two major components and two major phases. For two components, one is sentence reconstruction based on word set, and the other is back-translation. For two phases, they 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 use a word set constructor to construct the word set $WS = \{w_1, w_2, \cdots\}$. For the latter, we use a L_1-L_2 translation model to get the translated token sequence $Z = [z_1, z_2, \cdots]$ in L_2.

In the paraphrase generation phase, we employ a hybrid decoder with 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 should consider both content preservation and lexical variation.

For content preservation, we select informative words from the original sentence by either removing stopwords or retaining high-IDF words. Our choice here is removing stopwords, we will explain this later in Section 3.5. We stem the selected informative words to build the keywords set KWS, which will be passed to the next stage.

The keywords set represent the main semantics of the original sentence. Note that the same semantics can be represented by a variety of word sets. To increase the lexical diversity of the generated paraphrase, each word in KWS is randomly replaced with one of its synonyms[1], including itself. This process is known as "random replacement". We obtain WS after this step. BERT based methods can also be used to generate synonyms, we are not using them for two reasons: i) We have to generate synonyms for every single word in the training set, it is too computational expensive if we use BERT; 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 with no sequential information.

To train a set2seq model, we require the encoder to have no serialization processing for the input set. RNN-based models are inappropriate for set2seq due to their serialization nature, so 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.

[1]Synonyms are obtained from https://wordnet.princeton.edu/.

Algorithm 1 Paraphrasing Framework

Require: Original sentence $X = [x_1, x_2, \cdots]$;
Ensure: Paraphrase $Y = [y_1, y_2, \cdots]$;
1: Reduce X to a set of keywords KWS by removing stopwords;
2: Obtain WS from KWS by random replacement with synonyms;
3: Translate X into Language L_2: $Z = [z_1, z_2, \cdots]$;
4: Encode WS with set2seq to hidden state H_{ws};
5: Encode Z with L_2-L_1 translation model to hidden state H_{bt};
6: Initialize: Y = [], y_0 = BOS, t = 0;
7: **while** y_t != EOS and t <length-limit **do**
8: $t = t + 1$;
9: Calculate y_t with Eqn. 3;
10: Y.append(y_t);
11: **end while**
12: **return** Y;

We train set2seq with word set WS as the input and original sentence X as the output. Since we are not using any parallel data, the training process is considered unsupervised. 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}. Our purpose is to generate the output sequence $Y = [y_1, y_2, \cdots]$.

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:

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

Here P_{ws} and P_{bt} are the probabilities of v_i being next token calculated by the decoder of the set2seq model and the L_2-L_1 translation model respectively, and λ is the hyper-parameter to balance the weight between the two probabilities. Algorithm 1 shows the whole procedure of our paraphrasing framework.

3 Experimental Results

In this section, we first introduce the experimental setup, including dataset, baselines, evaluation metrics, and implementation details. Then, we show the results and compared with five unsupervised baselines and six supervised+domain-adapted baselines in Section 3.4. Finally, we analyze the result from four aspects: the effects of different datasets, ablation study, case study, and human evaluation.

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.

We randomly sample the remaining parallel paraphrases 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, 500K for WikiAnswers, 320K for MSCOCO and 110K for Twitter.

Quora. Quora[2] dataset is released by Quora in January 2017. It contains 400K pairs of questions with manual annotation about whether questions in each pair are duplicates of each other. Through these annotations, there are 140K pairs marked as paraphrases and 320K pairs masked as non-paraphrases.

WikiAnswers. WikiAnswers (Fader et al., 2013) dataset contains 2.3M pairs of question paraphrases extracted from the WikiAnswers website. The dataset is collected automatically without manual annotation.

MSCOCO. MSCOCO (Lin et al., 2014) contains human-annotated captions for 120K images. Each image contains five captions considered as paraphrases of each other, we take four pairs from each image and get 500K parallel pairs.

Twitter. Twitter (Lan et al., 2017) is a paraphrase detection dataset, containing 110K pairs of potential paraphrases and 60K manually annotated paraphrases. There are only 600 sentences marked as paraphrases in the test set, and we take them all for testing.

Training on Common-Domain Data When there is no sufficient target-domain non-parallel data, or when we cannot use any data from the target-domain to train the set2seq model, 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 Section 4 for data augmentation since we cannot train the set2seq model with the translation data to be augmented.

3.2 Baselines and Evaluation Metrics

We compare our framework with five unsupervised methods and six supervised methods with domain adaptation. We re-produce ParaNMT and ParaBank with our translation models, and take the results from Liu et al. (2019) for other baselines. For a fair comparison, we keep their scripts for data pre-processing and evaluation. On the Quora dataset, we even use the same train-test split as UPSA.[3]

Unsupervised methods. The current state-of-the-art unsupervised method is Unsupervised Paraphrasing by Simulated Annealing (UPSA), proposed by Liu et al. (2019), which is also our main target of comparison. Other unsupervised methods include CGMH from Miao et al. (2019), ParaNMT

[2]You can find the dataset at https://data.quora.com/First-Quora-Dataset-Release-Question-Pairs

[3]The pre-processing script, evaluation script, train-test split and results of UPSA can be found at https://github.com/anonymity-person/UPSA

from Wieting and Gimpel (2017), ParaBank(-3^{rd} IDF) from Hu et al. (2019b), and VAE from Kingma and Welling (2013). 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 shallow fusion from Gulcehre et al. (2015), Multi-Task Learning (MTL) from Domhan and Hieber (2017), Pointer-generator from See et al. (2017), Transformer (Vaswani et al., 2017) with copy mechanism, and MTL with copy mechanism.

Evaluation metrics. For the fairness of comparison, 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 NLP 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 lowercase 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. Same as UPSA and DNPG, all baselines include all words that appear in the training set into the vocabulary for a fair comparison. For the hyper-parameter λ mentioned in Section 2.4, we set it to 0.5 for all datasets(chosen from $\{0.1, 0.2, \cdots, 1, 2, 3\}$).

For the translation models in back-translation, we train them with the WMT17[4] zh-en dataset (Ziemski et al., 2016). We train them 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 mentioned in Section 3.1, 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. For some test cases, WikiAnswers does not have 5 references, so we evaluate them on reduced references. For ROUGE scores, we take the average of all references.

3.4 Results

Table 2 presents our experimental results. We mark the previous highest scores by underlining them and mark the present highest scores with the bold font. The supervised method (DNPG (SOTA)) here is only for reference.

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 useful information comes not only from translation, as the set2seq model alone can already outperform almost all competitors.

Our framework outperforms all existing unsupervised methods and supervised methods with domain adaptation. The results from our framework are even close to the state-of-the-art supervised model DNPG.

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. Sentences in Quora and WikiAnswers

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

		Quora				WikiAnswers			
	Model	iBLEU	BLEU	R-1	R-2	iBLEU	BLEU	R-1	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
	Shallow fusion	6.04	7.95	44.87	14.79	22.57	29.76	53.54	20.68
	MTL	4.90	6.37	37.64	11.83	18.34	23.65	48.19	17.53
	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	VAE	8.16	13.96	44.55	22.64	17.92	24.13	31.87	12.08
	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
	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
	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**

		MSCOCO				Twitter			
	Model	iBLEU	BLEU	Rouge1	Rouge2	iBLEU	BLEU	Rouge1	Rouge2
Unsupervised	VAE	7.48	11.09	31.78	8.66	2.92	3.46	15.13	3.40
	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
	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
	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 state-of-the-art method (Li et al., 2019) are only available on these two datasets.

are of the best quality. Experiments on these two datasets are the most persuasive and representative.

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. The set2seq-common model cannot learn the in-domain properties of MSCOCO, so it does relatively poorly here as opposed to its performance in other datasets.

Lack of training data for Twitter leads to insufficient training of most models. Models containing 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 outperforms retaining high-IDF words. For high-IDF words, we keep top $k\%$ high-IDF words in the original sentence. For the value of k, we set $k = 50$, which is the best among $[30, 40, 50, 60, 70]$. 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 lead to a high BLEU between source sentences and output paraphrases, which substantially reduces the diversity of generated sentences.

Case Study. Table 4 shows the examples of generated paraphrases through different strategies.

Two kinds of information are easily lost in set2seq: one is the information in stopwords; the other is the information in the sequential expression. In the first example, set2seq model loses the word "When" when generating paraphrase from

Model Variants	iBLEU	BLEU_{ref}	BLEU_{src}
set2seq+BT	**14.66**	22.53	**56.17**
⊖ excluding stopwords ⊕ retaining high-IDF	13.46	22.15	64.75
⊖ random replacement	13.78	**23.92**	77.47
⊕ position encoding	14.07	23.26	68.60

Table 3: Ablation Study on Quora.

Example 1	
Input	when will be end of world ?
Word Set	(stop, earth, ?)
BT	when is the end of the world ?
set2seq	will the world end ?
set2seq+BT	when will the world end ?
Example 2	
Input	could this universe be inside a black hole ?
Word Set	(universe, in, dark, cave, ?)
BT	can universe be a black hole ?
set2seq	is there a black hole in the universe ?
set2seq+BT	is the universe in a black hole ?
Example 3	
Input	do product ideas get seed fundings ?
Word Set	(produce, mind, incur, germ, financing, ?)
BT	does the product concept receive seed money ?
set2seq	where can i get funding for my product idea ?
set2seq+BT	how do i receive seed funding for my product idea ?

Table 4: Case Study

the word set. In the second example, set2seq model mistakes the relationship between the the universe and the black hole since it cannot obtain any sequential information.

For back-translation, the correct paraphrase sometimes cannot be generated due to the limited capacity of the translation models, "seed funding" should be a fixed phrase in Example 3, but back-translation cannot recognize it.

Method	Accuracy	Fluency
VAE	2.48	3.44
CGMH	2.97	3.67
UPSA	3.52	3.69
back-translation	3.50	**4.52**
set2seq+BT(ours)	**3.98**	4.43

Table 5: Results for Human Evaluation.

Human Evaluation. We choose 100 sentences from Quora and ask 3 human annotators to score the result from 1 to 5 from the perspective of both fluency and accuracy without telling them the result is generated by which method, the higher score indicates the better quality of the generated paraphrases. From the perspective of fluency, we judge whether paraphrase conforms to grammar and common sense. From the perspective of accuracy, we judge whether paraphrase has the same meaning as the original sentence but expressed differently.

We can see that word/phrase based methods have bad performances on fluency since their Language Model is trained on a small dataset. Paraphrases generated by back-translation are not very accurate since they are not trained by in-domain data. From the perspective of 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

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 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 (Section 3.1). 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, considering the length of sentences in the NMT training set is longer, we truncate sentences longer than 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 training data, we ob-

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

tain the de-en data from WMT17-europarl[5](Koehn, 2005), 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 50000 and extract vocabulary of up to 50000 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 6 shows the result. Paraphrase augmentation improves the model trained with original data pairs by anywhere from 1.53 to 2.17 BLEU.

In the process of producing paraphrase, we use some additional data, such as monolingual English data and Chinese English translation data. There may be other methods to improve the final results based on these data, but our purpose is to show the effectiveness of our methods in long sentences since only paraphrases with both accuracy and diversity can construct perfect translation pairs.

5 Related Work

We show the relevant work of paraphrase generation from the aspects of 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. Fu et al. (2019) also used BOW in their work, but they chose only a few important words to aid a seq2seq model, which is much different from ours. Other methods include VAE-SVG (Gupta et al., 2018) and transformer-pb (Wang et al., 2019), but these three methods perform worse than DNPG and have no discussion about domain adaptation, so we do not include them in our baselines.

For unsupervised methods, VAE(Kingma and Welling, 2013) can be used on this task directly, so it is often considered as one of the baselines. There are two mainstream approaches in recent works, one based on lexical expression and the other based on back-translation. For the former, 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. Unsupervised methods usually need common word-level knowledge to help them deal with the relationship between words, for these two methods, they used GloVE (Pennington et al., 2014), and for our method, we are using WordNet. We compare our framework with these two methods to show changes on the sentential level are more reliable than changes on the lexical level. For the latter, 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). We compare our framework with these two methods since they are also using back-translation.

6 Conclusion

In this paper, we proposed a novel framework for unsupervised paraphrase generation that outperforms most existing unsupervised methods and supervised methods with domain adaptation. While the results are positive, there still remains many problems to be studied. Can we find more underlying semantics? Is there a better evaluation metric than iBLEU? We plan to look into these questions in the future and generate better paraphrases.

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

References

Tobias Domhan and Felix Hieber. 2017. Using target-side monolingual data for neural machine translation through multi-task learning. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1500–1505.

Sergey Edunov, Myle Ott, Michael Auli, and David Grangier. 2018. Understanding back-translation at scale. *arXiv preprint arXiv:1808.09381*.

Anthony Fader, Luke Zettlemoyer, and Oren Etzioni. 2013. Paraphrase-driven learning for open question answering. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1608–1618.

Yao Fu, Yansong Feng, and John P Cunningham. 2019. Paraphrase generation with latent bag of words. In *Advances in Neural Information Processing Systems*, pages 13623–13634.

Juri Ganitkevitch, Benjamin Van Durme, and Chris Callison-Burch. 2013. Ppdb: The paraphrase database. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 758–764.

Caglar Gulcehre, Orhan Firat, Kelvin Xu, Kyunghyun Cho, Loic Barrault, Huei-Chi Lin, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2015. On using monolingual corpora in neural machine translation. *arXiv preprint arXiv:1503.03535*.

Ankush Gupta, Arvind Agarwal, Prawaan Singh, and Piyush Rai. 2018. A deep generative framework for paraphrase generation. In *Thirty-Second AAAI Conference on Artificial Intelligence*.

J Edward Hu, Huda Khayrallah, Ryan Culkin, Patrick Xia, Tongfei Chen, Matt Post, and Benjamin Van Durme. 2019a. Improved lexically constrained decoding for translation and monolingual rewriting. 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 839–850.

J Edward Hu, Rachel Rudinger, Matt Post, and Benjamin Van Durme. 2019b. Parabank: Monolingual bitext generation and sentential paraphrasing via lexically-constrained neural machine translation. *arXiv preprint arXiv:1901.03644*.

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

Diederik P Kingma and Max Welling. 2013. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*.

Philipp Koehn. 2005. Europarl: A parallel corpus for statistical machine translation. In *MT summit*, volume 5, pages 79–86. Citeseer.

Wuwei Lan, Siyu Qiu, Hua He, and Wei Xu. 2017. A continuously growing dataset of sentential paraphrases. *arXiv preprint arXiv:1708.00391*.

Zichao Li, Xin Jiang, Lifeng Shang, and Qun Liu. 2019. Decomposable neural paraphrase generation. *arXiv preprint arXiv:1906.09741*.

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

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

Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. 2014. Microsoft coco: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer.

Xianggen Liu, Lili Mou, Fandong Meng, Hao Zhou, Jie Zhou, and Sen Song. 2019. Unsupervised paraphrasing by simulated annealing. *arXiv preprint arXiv:1909.03588*.

Ning Miao, Hao Zhou, Lili Mou, Rui Yan, and Lei Li. 2019. Cgmh: Constrained sentence generation by metropolis-hastings sampling. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 6834–6842.

Rada Mihalcea and Paul Tarau. 2004. Textrank: Bringing order into text. In *Proceedings of the 2004 conference on empirical methods in natural language processing*, pages 404–411.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting on association for computational linguistics*, pages 311–318. Association for Computational Linguistics.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543.

Aaditya Prakash, Sadid A Hasan, Kathy Lee, Vivek Datla, Ashequl Qadir, Joey Liu, and Oladimeji Farri. 2016. Neural paraphrase generation with stacked residual lstm networks. *arXiv preprint arXiv:1610.03098*.

Abigail See, Peter J Liu, and Christopher D Manning. 2017. Get to the point: Summarization with pointer-generator networks. *arXiv preprint arXiv:1704.04368*.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Edinburgh neural machine translation systems for wmt 16. *arXiv preprint arXiv:1606.02891*.

Hong Sun and Ming Zhou. 2012. Joint learning of a dual smt system for paraphrase generation. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Short Papers-Volume 2*, pages 38–42. Association for Computational Linguistics.

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.

Su Wang, Rahul Gupta, Nancy Chang, and Jason Baldridge. 2019. A task in a suit and a tie: paraphrase generation with semantic augmentation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 7176–7183.

John Wieting and Kevin Gimpel. 2017. Paranmt-50m: Pushing the limits of paraphrastic sentence embeddings with millions of machine translations. *arXiv preprint arXiv:1711.05732*.

Michał Ziemski, Marcin Junczys-Dowmunt, and Bruno Pouliquen. 2016. The united nations parallel corpus v1. 0. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 3530–3534.