Unsupervised Paraphrasing via Sentence Reconstruction and Back-translation

Anonymous ACL 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.

Domain adaptation is a common requirement in supervised paraphrase generation since 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,

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.

and Twitter (Lan et al., 2017), and achieve the state-of-the-art accuracies compared to existing unsupervised 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 cross-domain dataset and test it on these four datasets, and still obtain decent results. We call the set2seq model trained from the big cross-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 argument 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

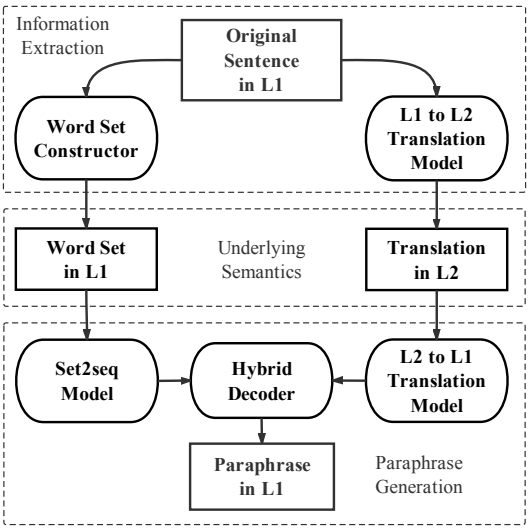

Figure 1: Our Paraphrsing Framework

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.

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

Input: Original sentence $X = [x_1, x_2, \cdots]$;
Output: 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 influence of dataset, 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 Cross-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 cross-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.

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 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. Li et al. (2019) raised the issue of domain adaptation in his paper and demonstrated that DNPG

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

also performed best with domain adaptation, so we mainly compare our framework with DNPG. 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 [3], which are iBLEU (Sun and Zhou, 2012), BLEU (Papineni et al., 2002) and ROUGE (Lin, 2004) scores. BLEU and ROUGE scores are common evaluation matrics for NLP tasks while iBLEU is especially designed for paraphrase generation tasks. It penalizes 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.

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 50K vocabulary for all models. For baselines using back-translation (ParaNMT and ParaBank), we use the same vocabulary. For other baselines, we 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.

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 Tesla V100 GPUs. 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 the same transformer as in the translation models. For the domain-specific set2seq models, we use a 2-layer transformer with 300 embedding size, 4 heads, 1024 feed-forward dimensions, AdamOptimizer, and 0.1 dropout for all layers to train them. The training lasts 3 hours on a single Tesla V100 GPU for each dataset.

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 if there exists more than one reference.

3.4 Results

Table 2 presents our experimental results. For all evaluation metrics, a higher score represents better performance. 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 since it is an unfair comparison between supervised and unsupervised methods.

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 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.

[3] The evaluation script can be found at https://github.com/anonymity-person/UPSA

[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)	10.76	15.84	52.34	25.18	14.71	19.73	30.34	9.91
	ParaBank	9.86	14.61	49.90	23.71	13.08	17.47	28.89	9.32
	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)	13.67	20.57	58.33	32.54	26.26	33.74	56.16	23.12
	set2seq-common+BT (ours)	12.52	18.74	57.09	31.13	24.98	33.36	55.75	23.03
	set2seq+BT (ours)	**14.65**	**22.43**	**59.94**	**34.02**	**28.31**	**37.47**	**56.82**	**24.91**

		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)	7.48	10.83	30.89	8.70	7.60	10.83	35.45	14.78
	ParaBank	6.43	9.44	29.12	8.25	6.55	9.79	34.51	13.94
	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)	**11.51**	17.52	39.75	13.66	5.77	7.56	31.63	10.97
	set2seq-common+BT (ours)	8.99	13.31	35.82	11.03	9.71	**14.25**	**39.14**	**18.77**
	set2seq+BT (ours)	11.37	**17.91**	**40.27**	**14.12**	**9.86**	13.88	39.09	18.15

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.

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 ablation study on 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 $[30, 40, 50, 60, 70]$.

Model Variants	iBLEU	BLEU_{ref}	BLEU_{src}
set2seq+BT	**14.65**	22.43	**55.40**
⊖ excluding stopwords ⊕ retaining high-IDF	13.63	22.39	65.21
⊖ random replacement	13.83	**23.82**	76.08
⊕ position encoding	14.21	23.43	68.77

Table 3: Ablation Study on Quora.

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

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 ?
seq2seq+BT	what is the end of the world ?
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 ?
seq2seq+BT	could the universe be in a black hole ?
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 ?
seq2seq+BT	do product ideas get seed funding ?
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 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.

For seq2seq+BT, the generated sentences are too close to the original sentences by the order of the words. Our goal is to generate sentence level paraphrases, but seq2seq model limits the sequential expression.

Method	score
set2seq	2.79
back-translation	3.02
seq2seq+BT	3.34
set2seq+BT	3.57

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, the higher score indicates the better quality of the generated paraphrases. All annotators are asked to consider the result from both accuracy and diversity. We give a reference for scoring:

1. The meaning is totally different.
2. Exactly the same sentence.
3. The meaning is slightly different.
4. Express the same meaning in a slightly different expression.
5. Express the same meaning in a totally different expression.

Table 5 shows the average rating of all annotators on all sentences. Our framework performs the best among different baselines.

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.

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 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, which is 20 times the original 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 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 obtain the de-en data from WMT17-europarl[5](Koehn,

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

	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.

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. The result demonstrates that our framework can generate high-quality paraphrases 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 and Question Answering (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, as well as supervised methods with domain adaptation. While the results are positive, there still remains many problems to be studied. Can we extract the word set with a seq2set model instead of rule-based methods? Can we find more underlying semantics? Is there a better evaluation metric than iBLEU to balance accuracy and diversity? We plan to look into these questions in the future and generate better paraphrases.

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.

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.

Reviews

MetaReview
Comments:
AC#1: An unsupervised paraphrase generation approach based on a hybrid decoder that combines outputs of two unsupervised paraphrasing models; one is, a set-to-sequence model that generates sentences from a set of content words (synonyms+content words from the original input sentence), the other is a back-translation -based paraphrasing model. The approach is simple (though a bit incremental), exploits existing resources/corpora/models and shows good results in automatic metrics. The paper is well written, very clear exposition of the approach. Evaluation is done on several relevant well-known benchmarks and includes sota systems. A weak point is the human evaluation which is rather limited; only variants of the proposed models are evaluated instead of comparing against more competitive comparison systems.
AC#2: 1) The models are heavily trained (multiple V100s for days) and there is no mention at all how their baselines are trained. What data was the VAE trained? How much ParaNMT/ParaBank data was used? What architectures were used? There is no information about these baselines and I am skeptical these were adequately tuned based on results in Table 2. For instance, the set2seq is so much better when it completely ignores word order...part of this is possibly due to automatic metrics evidenced by how in human evals set2seq performs much worse than other approaches but better in Table 2. This largely invalidates Table 2 in my opinion as this model is completely unaware of word order and should not be able to generate paraphrases for complex sentences in contrast to the baselines. Note that these datasets are primarily based on short sentences... 2) Their human evals are only on ablations of their model - what about comparisons to other models? This would be especially true since Table 2 isn't all that reliable as seen by my earlier comment. 3) Using a bag of words is not a new idea for generation as a way to represent content and this paper is just adding this to round-trip backtranslation, combining two ideas. Though to be fair, using a bag-of-words in this exact way is new as far as I can tell and a quick lit search. 4) Data augmentation for MT is flawed to me because their gains are probably largely due to having more English data. Since they are translating X->En having a better decoder helps a lot, especially in low resource settings. Therefore adding 20x more English data alone would be helpful whether paraphrases or not. Notice that no baselines were used for this experiment as well.

Review #1
What is this paper about, what contributions does it make, what are the main strengths and weaknesses?
This work proposes a system for unsupervised paraphrase generation. The system consists two components, a back-translation model and a keyword-to-text generation model. The back-translation model generates a paraphrase by pivot translation; while the keyword-to-text generation model, which is essentially a set2seq model. The keywords are extracted with heuristics-based method and augmented by their synonyms. The final paraphrase is generated via the mixture of the prediction of decoders in translation and set2seq model. Extensive experiments has been done in multiple datasets to show the superior performance of the proposed method than other baseline models. This presented method is quite effective, but it is also very computational expensive.

Reasons to accept
The proposed method is simple and effective, which is a big plus for the task of paraphrase generation / data augmentation.
The experiments done in this work are very thorough and solid. The ablation study clearly shows the contribution of each component.

Reasons to reject
The most significant problem is the heavy usage of computation resources. There are totally three encoder-decoder networks in the proposed system. This make the method a little bit too incremental.
It would be better to include the performance of the strongest baseline in the data augmentation section.
This method still relies on at least two large-scale and high-quality parallel dataset for two well-trained NMT models; while some other baseline model like UPSA and VAE do not need them.

I am not sure whether the keyword-to-text method can still work when the source sentence are long. In that case, seems that there would be significant information loss with only a set of words. All of the datasets employed by this paper consists short sentences.

Overall Recommendation: 3.5

Review #2
What is this paper about, what contributions does it make, what are the main strengths and weaknesses?
The paper suggests a hybrid decoder to introduce diversity in automatically-generated paraphrases.
One path to generate paraphrase is round-trip translation; an alternate path is to reduce words to a selected word set, perturb those words, then generate some a sentence from this bag. The perturbation is done using wordnet synonyms, a potentially limited set of operations.
The authors evaluate on a set of paraphrase tasks and compare against many baselines -- good evaluation.
The translation system is only one sample and not large -- a Chinese-English system trained on WMT data -- and yet the authors make claims about back-translation as a whole. This may be misleading.
Adding diversity through WordNet synonyms seems like a good baseline, but I'm concerned about the converage of WordNet (what's the OOV rate on each of these sets?) as well as the accuracy of the synonym replacements. I wish there were more experiments about the qualitative and quantitative impact here.

Reasons to accept
Good thinking about how to generate interesting paraphrases.
Good evaluation setup, with many reasonable baselines.
Some human evaluation, although limited.
Evaluation in MT setting.

Reasons to reject
The authors group all back translation methods together, and use a potentially challenging language pair (Chinese-English). At least one other paper has found back translation with Chinese-English produces lesser quality paraphrases when compared to other language pairs: https://www.aclweb.org/anthology/D19-5503/
The word set method seems appropriate for shorter sentences, but may struggle to retain semantics of longer sentences.

Overall Recommendation: 3.5

Questions for the Authors(s)
What's the quality of your English-Chinese machine translation set, compared to the state-of-the-art?
What are the out-of-vocabulary rates for words in WordNet? That is, given a word set, what percentage of the tokens have synonynms?
Have you considered some "bias" -- not replace every word, but replace words at random, and change the random threshold?
Have you considered alternate methods for generating paraphrase words, like using BERT but masking the token and letting BERT suggest replacements? Or sampling within a distance given GloVe embeddings? Something that is smoother than synsets?
What is the value of alpha in iBLEU? Could you include that?

Typos, Grammar, and Style
Will follow up later.

Review #3
What is this paper about, what contributions does it make, what are the main strengths and weaknesses?

This paper introduces a new ways to train paraphrase generation model. This works combines popular back-translation methodology with word-set to sentence model using a common decoder. The word-set to sentence model is trained in "unsupervised" manner through creating an input by taking a sentence and removing stop words and randomly replacing each word with its synonyms and training the model to reconstruct the original sentence. The paper presents comparison against many different paraphrase generation methodologies as well as thorough ablation studies, human evaluation, and evaluation of generated paraphrase using an external task (using them to augment machine translation training data.)
The presented methodology is simple yet effective and the presented experiments are convincing. However, it lacks analysis on cases where removing stop words impacts semantics (negative polarity) and the method cannot be applicable to languages without resources like wordnet.

Reasons to accept
The new method introduced method for unsupervised paraphrase generation is simple yet effective. People who are building paraphrase models can immediately benefit from the presented method. The paper is reasonably clearly presented.

Reasons to reject
Some may view the newly introduced methodology uninspiring and perhaps the paper needs to be more thorough such as exploring different "perturbation" methods and different aspects of domain adaptation.

Overall Recommendation: 3.5

Questions for the Authors(s)
how does removing some stop words impacts output? (e.g., words such as "not") Does the back-translation model prevents such cases from reversing the polarity? what were the exact list of stop words that were removed from word sets?
have you tried different word perturbation methods? (more model-based methods and not based on heuristics)
"KWS is randomly replaced with one of its synonyms, including itself" -> some clarification is needed. Does this mean that every word is replaced? or does "including itself" mean that some word stay the same?

Missing References
In the first paragraph in the introduction, authors state that "such as question answering, machine translation, and information retrieval." It would be nice if citations can be added for how paraphrase generations helps each use cases.

Typos, Grammar, and Style
last paragraph in the introduction: "our method to argument" -> augment. in "evaluation metrics" paragraph: "evaluation matrics" -> metrics

Response

reviewer 1

Q: About computational expense. A: This is a good point, let us analyze it from two stages. For the training stage, we can reuse the translation model for any given domain, or even take one pretrained translator from the open-source codes. The set2seq is actually lighter, as it only has 2 layers for encoder and decoder. For the testing stage, the cost for set2seq can be ignored, and the majority of the cost comes from machine translation, which could be our future research.

Q: Strongest baseline in the data augmentation. A: The section "Application on Translation Task" is there to demonstrate one possible downstream task and our system's capability to deal with long

sentences (translation data contains longer sentences). This can be a new research topic left for future works.

Q: Relies on at least two large scale parallel datasets. A: It is true we need parallel data, but we only need one parallel dataset (reuse it for both sides translation) with a commonly used target-language, which is easy to find. Besides, set2seq itself doesn't need any parallel data, and it already outperforms other baselines.

reviewer 2

Q: Why are we using Chinese in back-translation? A: We are very grateful that you can point this out. We don't have strict requirements for the language used in back-translation, so we just use a commonly used translation pair. We will analyze the effect of different languages in future works. Q: Quality for back-translation data. A: For translation tasks, we are using a standard transformer. For Chinese->English, our BLEU: 24.2, SOTA: 27.2. For English->Chinese, our BLEU: 34.1, SOTA: 36.3.

Q: OOV rate and accuracy for WordNet. A: OOV rate: 29.1%, accuracy: 45.38% (take 100 sentences, 412 words can be replaced, 187 correct replacements)

Q: Alpha in iBLEU. A: The value for alpha is 0.9, the same as UPSA and DNPG. We will make this clear in our paper.

reviewer 3

Q: Languages without WordNet. A: This is a good concern. We use WordNet to find synonyms here, but there are other alternatives like BERT, GloVe embeddings or other tools to find synonyms, such as "BabelNet", which is the multilingual version of WordNet containing 271 languages.

Q: The impact of removing stopwords like "not". A: That's true, and it's one of the problems with set2seq model. When we remove "not" as a stopword, set2seq itself cannot realize there should be a "not". However, when the back-translation comes in, the hybrid model can realize that. This is why we are using different "underlying semantics": to complement each other. For stopwords, we use the list provided by nltk.

General

Our sincere gratitude for all your valuable comments and suggestions!

Regarding the replacement of words: We are not using a random threshold to decide whether a word should be replaced. Instead, we add the word itself into the synonym set, so it has some chance to be replaced by itself (that is, remain unchanged). We are doing this to prevent the set2seq model from learning that the word in word set "must" be replaced.

The idea that using BERT or GloVe to generate synonyms is really good, they can generate more smooth and accurate synonyms. There is a balance here between accuracy and diversity, we need more experiments to see whether "accurate" synonyms like BERT or "diverse" synonyms like WordNet are better. We are focusing on the thought of combine two "underlying semantics" together in this paper, but this idea could be one of our next research topics in future work.

For the problem of long sentences, the hybrid decoder is the result of both the set2seq and the back-translation. They work together to generate paraphrases. The set2seq itself may have bad performances when it comes to long sentences, but the hybrid decoder still has good performances. We don't have a benchmark dataset to test the result on long sentences, but we show our model's capability to deal with long sentences by augmenting data of NMT tasks. The average length of training data of set2seq-common is 22.4 tokens per sentence while the average length of the NMT data (data to be augmented) is 28.15 tokens per sentence, we consider them as

"long sentences". The generated paraphrases should be in good quality since they can improve the result of NMT. If this is still not convincing enough, here are 2 examples from the NMT data generated with set2seq-common+BT:

original: as has already been said , thousands of refugees have fled from darfur to chad , but with continuing violence in chad people are now also escaping to cameroon , nigeria and the central african republic . paraphrase: thousands of refugees have fled from darfur to chad , as has already been said , but the continued violence in chad has also led to the flee of people to cameroon , nigeria and the central african republic .

original: therefore , the committee on women 's rights and equal opportunities proposes that parliament should adopt a plan of action alongside measures which , of course , are provided for in the legislation and are aimed at reaching the objectives we wish to achieve . paraphrase: the women 's rights and equal opportunities commission , therefore , has proposed that the parliament adopt an action plan and that it adopt measures which are , of course , provided for by legislation aimed at achieving the objectives we aim to achieve .

Summary & Changes

Average score in ACL: 3.5
Main reason to reject: lack of human evaluation & computational expensive.
Our improvement:

1. We added human evaluation for all unsupervised baselines and evaluate them from the perspective of both fluency and accuracy. (**Page 7, Section 3.5, Human Evaluation).**
2. We moved all our experiments to GTX-2080 GPUs to show we are not relying on V100. We also showed that we use our translation models to re-produce ParaNMT and ParaBank. What's more, we showed that set2seq itself is not computationally expensive. We also decreased the size of vocabulary from 50k to 30k. (**Page 5, Section 3.3**)
   We changed the results in **(Page 6, Table 2)** based on the new experiments**,** actually, they are similar to the results in the previous version.
   Here are some explanations:
   1) We moved the training process from V100 to GTX-2080. Why does it take the same time? Take the translation models as an example, the training on the V100 converged in less than 3 days, but we still kept it running for a while to ensure convergence. When training on GTX-2080, we stopped training in time as soon as the model converged.
   2) The translation model is still "computationally expensive".
      As a translation model, it cannot be regarded as a large model, which only requires two GTX-2080 GPUs. Many laboratories can train such a model and reuse it for different tasks.
      We divide all methods into two categories: need back-translation and don't need back-translation, and the internal model complexity of these two categories is similar.
      a) methods with back-translation: set2seq+BT, ParaNMT, ParaBank. They all use the same translation models.
      b) methods without back-translation: set2seq only, UPSA, CGMH, VAE
         i. set2seq uses a 2-layer transformer
         ii. UPSA and CGMH both use a 2-layer LSTM to train a Language Model
         iii. VAE uses a 2-layer LSTM seq2seq model

      They all have similar sizes of parameters. Actually, both UPSA and CGMH use the pre-trained GloVe embedding, which is trained from a large corpus for a long time.
3. Some reviewers asked us why we are not using BERT when generating synonyms. We explained it this time. (**Page 3, Section 2.2**)
4. The author of UPSA showed its test results of the Quora dataset on his Github, so we followed the train-test split of UPSA on the Quora dataset for a fair comparison. **(Page 4, Section 3.2)**