Paraphrase Generation as Unsupervised Machine Translation

Xiaofei Sun[1,2], **Yufei Tian**[3], **Yuxian Meng**[2], **Nanyun Peng**[3]
Fei Wu[1,5], **Jiwei Li**[1,2] **and Chun Fan**[4]
[1]Zhejiang University, [2]Shannon.AI, [3]University of California, Los Angeles
[4]Peng Cheng Laboratory, [4]National Biomedical Imaging Center, Peking University
[4] Computer Center, Peking University, [5]Shanghai AI Laboratory
[5]Shanghai Institute for Advanced Study of Zhejiang University
xiaofei_sun@zju.edu.cn, yufeit@ucla.edu, jiwei_li@shannonai.com

Abstract

In this paper, we propose a new paradigm for paraphrase generation by treating the task as unsupervised machine translation (UMT) based on the assumption that there must be pairs of sentences expressing the same meaning in a large-scale unlabeled monolingual corpus. The proposed paradigm first splits a large unlabeled corpus into multiple clusters, and trains multiple UMT models using pairs of these clusters. Then based on the paraphrase pairs produced by these UMT models, a unified surrogate model can be trained to serve as the final SEQ2SEQ model to generate paraphrases, which can be directly used for test in the unsupervised setup, or be finetuned on labeled datasets in the supervised setup. The proposed method offers merits over machine-translation-based paraphrase generation methods, as it avoids reliance on bilingual sentence pairs. It also allows human intervene with the model so that more diverse paraphrases can be generated using different filtering criteria. Extensive experiments on existing paraphrase dataset for both the supervised and unsupervised setups demonstrate the effectiveness the proposed paradigm.

1 Introduction

The goal of paraphrase generation (Prakash et al., 2016a; Cao et al., 2016; Ma et al., 2018; Wang et al., 2018) is to generate a sentence semantically identical to a given input sentence but with variations in lexicon or syntax. It has been applied to various downstream NLP tasks such as parsing (Berant and Liang, 2014), question answering (Dong et al., 2017), summarization (Barzilay, 2004) and machine translation (Callison-Burch et al., 2006).

Building a strong paraphrase generation system usually requires massive amounts of high-quality annotated paraphrase pairs, but existing labeled datasets (Lin et al., 2014; Fader et al., 2013; Lan et al., 2017) are either of small sizes or restricted in narrow domains. To avoid such a heavy reliance on labeled datasets, recent works have explored unsupervised methods (Li et al., 2018b; Fu et al., 2019; Siddique et al., 2020) to generate paraphrase without annotated training data, among which the back-translation based model is an archetype (Mallinson et al., 2017; Sokolov and Filimonov, 2020). It borrows the idea of back-translation (BT) in machine translation (Sennrich et al., 2016) where the model first translates a sentence s_1 into another sentence s_2 in a different language (e.g., En→Fr), and then translates s_2 back to s_1. In this way, the model is able to generate paraphrases by harnessing bilingual datasets, removing the need for label paraphrase data.

However, BT-based models for paraphrase generation have the following severe issues: firstly, BT-based systems rely on external resources, i.e., bilingual datasets, making them hard to be applied to languages whose bilingual datasets are hard to obtain. Secondly, translation errors, such as duplicate words (Holtzman et al., 2020), missing words (Luong et al., 2015) and polysemous words (Rios Gonzales et al., 2017), will accumulate during the forward and backward translations, resulting in inferior performances. Thirdly, machine translation models work like blackboxs, making it hard for humans to intervene with the model and control the generation process.

In this work, we propose to address these problems based on the assumption that there must be pairs of sentences expressing the same meaning in a large-scale unlabeled corpus. Inspired by unsupervised machine translation (UMT) models, which align semantic spaces of two languages using monolingual data, we propose a pipeline system to generate paraphrases following two stages: (1) splitting a large-scale monolingual corpus into multiple clusters/sub-datasets, on which UMT models are trained based on pairs of these sub-datasets; and (2) training a unified surrogate model based on the paraphrase pairs produced by the trained multiple

Proceedings of the 29th International Conference on Computational Linguistics, pages 6379–6391
October 12–17, 2022.

UMT models, where we can design filtering functions to remove the pairs with undesired properties. The unified surrogate model can be then directly used for paraphrase generation in the unsupervised setup, or be finetuned on labeled datasets in the supervised setup.

The proposed framework provides the following merits over existing BT-based methods: (1) it is purely based on a large-scale monolingual corpus, which removes the reliance on bilingual datasets; (2) the trained unified model is able to generate paraphrases end-to-end, which avoids the issue of error accumulation that exists in vanilla BT-based models; and (3) human interventions can take place in the filtering step, which gives finer-grained controls over the generated paraphrases.

We conduct extensive experiments on a wide range of paraphrase datasets to evaluate the effectiveness of the proposed framework, and we are able to observe performance boosts against strong baselines in both supervised and unsupervised setups.

2 Related Work

Paraphrase Generation Methods for paraphrase generation usually fall into two categories: supervised and unsupervised approaches. Supervised methods for paraphrase generation rely on annotated paraphrase pairs. Xu et al. (2018); Qian et al. (2019) employed distinct semantic style embeddings to generate diverse paraphrases, and Iyyer et al. (2018); Li et al. (2019); Chen et al. (2019); Goyal and Durrett (2020) proposed to use different syntactic structure templates. A line of work (Mallinson et al., 2017; Sokolov and Filimonov, 2020) formalized paraphrase generation as machine translation. Unsupervised paraphrase generation is primarily based on reinforcement learning (RL) generative models (Ranzato et al., 2015; Li et al., 2016b). RL optimizes certain criteria, e.g. BLEU, to reward paraphrases with higher quality (Li et al., 2018b; Siddique et al., 2020). Bowman et al. (2016); Yang et al. (2019) trained a variational auto-encoder (VAE) (Kingma and Welling, 2013) to generate paraphrases. Other unsupervised methods for paraphrase generation include VAE (VQ-VAE) (Roy and Grangier, 2019), latent bag-of-words alignment (Fu et al., 2019) and simulated annealing (Liu et al., 2019a). Adapting large-scale pretraining (Devlin et al., 2018; Radford et al., 2018; Liu et al., 2019b; Clark et al., 2020; Sun et al., 2021b) to paraphrase generation has been recently investigated (Witteveen and Andrews, 2019; Hegde and Patil, 2020; Niu et al., 2020; Meng et al., 2021) and has shown promising potentials to improve generation quality. Our work is distantly related to unsupervised text style transfer (Hu et al., 2017; Mueller et al., 2017; Shen et al., 2017; Li et al., 2018a; Fu et al., 2018), where the model alters a specific text attribute of an input sentence (such as sentiment) while preserving other text attributes.

Regarding soliciting large-scale paraphrase datasets, Bannard and Callison-Burch (2005) used statistical machine translation methods obtain paraphrases in parallel text, the technique of which is scaled up by Ganitkevitch et al. (2013) to produce the Paraphrase Database (PPDB). Wieting et al. (2017) translate the non-English side of parallel text to obtain paraphrase pairs. Wieting and Gimpel (2017) collected paraphrase dataset with million of pairs via machine translation. Hu et al. (2019a,b) produced paraphrases from a bilingual corpus based on the techniques of negative constraints and inference sampling.

Unsupervised Machine Translation Unsupervised Machine Translation(UMT) has been an active research direction in NLP (Ravi and Knight, 2011). Pioneering work for unsupervised neural machine translation used denoising auto-encoders and back-translation (Sennrich et al., 2016) to iteratively refine the generated translation. Artetxe et al. (2017) proposed to use a shared encoder to encode source input sentences from different languages. Lample et al. (2017) additionally used adversarial and cross-domain training objectives to better identify different language domains. Yang et al. (2018) relaxed the strategy of sharing the entire encoder in Artetxe et al. (2017) by building independent encoders to maintain unique characteristics of each language. Another line of work for UMT is to combine statistical machine translation (SMT) and NMT. Artetxe et al. (2018); Lample et al. (2018) built a phrase-level mapping table from the source language to the target language. Following works improved UMT by combining SMT and NMT in different ways, such as warming up an NMT model with a trained SMT model (Marie and Fujita, 2018; Artetxe et al., 2019) and using SMT as posterior regularization (Ren et al., 2019). Other works involve initializing the model using retrieved semantically similar sentence pairs (Wu et al., 2019; Ren et al., 2020; Sun et al., 2021a), using auxiliary parallel data (Li et al., 2020; Garcia et al., 2020) and pretraining on large-scale multi-lingual data (Lample and Conneau, 2019; Song et al., 2019; Liu et al., 2020; Zhu et al., 2020).

3 Background for Unsupervised Machine Translation

We use the unsupervised machine translation (UMT) framework proposed by Lample et al. (2017) as the backbone. We briefly go though the model structure in this section. Let C_{src} and C_{tgt} respectively denote the monolingual dataset for the source and target language, on which a translation model M is trained to to generate target sequences y based on source sequences x, $y = M(x)$. The model is first initialized by training in a word-by-word translation manner using a parallel dictionary. The initial parallel dictionary is thus a word being translated to itself. Next, the model is iteratively trained based on a denoising auto-encoding (DAE), back-training (BT) and adversarial learning (AL). DAE allows the model to reconstruct the translation from a noisy input sentence by dropping and swapping words in the original sentence. The training objective of DAE is given by:

$$\mathcal{L}^{l}_{\text{DAE}} = \mathbb{E}_{x\sim C_l, \hat{x}\sim d(e(N(x),l),l)}[\Delta(\hat{x}, x)] \quad (1)$$

where $l = src$ or $l = tgt$ specifies the language, $N(x)$ is a noisy version of x, e and d respectively means encoding and decoding, and Δ measures the difference between the two sequences, which is the cross-entropy loss in this case. BT encourages the model to reconstruct the input sentence x from $N(y)$, a corrupted version of the model's translation $y = M(x)$. The training objective is given by:

$$\mathcal{L}^{l_1\to l_2}_{\text{BT}} = \mathbb{E}_{x\sim C_{l_1}, \hat{x}\sim d(e(N(M(x)),l_2),l_1)}[\Delta(\hat{x}, x)] \quad (2)$$

AL uses a discriminator to distinguish the language from the encoded latent representations, and by doing so, the model is able to better map two languages into the same latent space. The discriminative training objective is given by:

$$\mathcal{L}^{l}_{\text{Dis}} = -\mathbb{E}_{(x,l)}[\log p(l|e(x,l))] \quad (3)$$

The encoder is trained to fool the discriminator so that the encoder and the discriminator perform together in an adversarial style (Goodfellow et al., 2014):

$$\mathcal{L}^{l_1\to l_2}_{\text{Adv}} = -\mathbb{E}_{(x_1,l_1)}[\log p(l_2|e(x_1,l_1))] \quad (4)$$

The final training objective is given by:

$$\begin{aligned}\mathcal{L} = \lambda_1[\mathcal{L}^{l_1}_{\text{DAE}} + \mathcal{L}^{l_2}_{\text{DAE}}] + \lambda_2[\mathcal{L}^{l_1\to l_2}_{\text{BT}} + \mathcal{L}^{l_2\to l_1}_{\text{BT}}] \\ + \lambda_3[\mathcal{L}^{l_1\to l_2}_{\text{Adv}} + \mathcal{L}^{l_2\to l_1}_{\text{Adv}}]\end{aligned} \quad (5)$$

The discriminative loss $\mathcal{L}^{l}_{\text{Dis}}$ is alternatively optimized with $\mathcal{L}$ to train the discriminator. We follow Lample et al. (2017) to implement each of the UMT models. We used the transformer-large (Vaswani et al., 2017) as the backbone instead of LSTMs in Bahdanau et al. (2014).

4 Model

The core idea of the proposed strategy is to use two subdatasets from a large monolingual corpus C and train unsupervised NMT models based on the two subdatasets. The path towards this goal naturally constitutes two modules: (1) constructing two subdatasets C_{src} and C_{tgt} from C; and (2) training the UMT model based on C_{src} and C_{tgt}.

4.1 Dataset Split

The crucial part in the framework is how to build the two subdatasets, on which the unsupervised NMT model is trained. To this end, we propose to (1) first construct candidates $\{c_1, c_2, ..., c_K\}$ for C_{src} and C_{tgt} from C based on the clustering models; and (2) selecting C_{src} and C_{tgt}. Based on C_{src} and C_{tgt}, UMT models will be trained. We use two criteria for clustering, LDA (Blei et al., 2003) and K-means clustering. The number of clusters/topics K is set to 80.[1]

LDA Clustering For LDA, we use Gibbs sampling and iterate over the entire corpus 5 times in total. In the last round, a sentence is assigned to the topic/cluster which has the largest probability of generating it. In LDA, each cluster is characterized as a distribution over the vocabulary. The distance between two subset c_m, c_n is the Jensen–Shannon (JS) divergence between the two distributions over the vocabulary:

$$\begin{aligned}\text{Dis}(c_m, c_n) &= \text{KL}(c_m||c_n) + \text{KL}(c_n||c_m) \\ \text{KL}(c_m||c_n) &= -\sum_{v\in V} p(v|c_m)\log\frac{p(v|c_m)}{p(v|c_n)} \\ \text{KL}(c_n||c_m) &= -\sum_{v\in V} p(v|c_n)\log\frac{p(v|c_n)}{p(v|c_m)}\end{aligned} \quad (6)$$

Since topics clustered by LDA can be incoherent (e.g., the clustering of stop words), we ask humans

[1] Here we use "topic" and "cluster" interchangeably.

to examine the top words of the topics, and discard meaningless clusters.

K-means Clustering For K-means, we use the average of the top layer embeddings from BERT (Devlin et al., 2018) to represent the sentence. Let h_s denote the sentence representation for the sentence s. We run the hard K-means model on the corpus, where the distance between a sentence and the cluster center is the L_2 distance between the two vector representations.

The LDA and K-means methods described above focus more on the situation that centers of two clusters are far away, but not individual sentences belonging to different clusters are different. These two focuses are correlated, but not exactly the same. The JS divergence for LDA clusters and L_2 distance for K-means clusters will be updated after the post-processing stage. LDA and K-means algorithms are performed on part of the the CommonCrawl corpus containing 10 billion English tokens.

4.2 UMT Training on C_{src} and C_{tgt}

4.2.1 Multiple UMT Models

We can randomly pick one pair of subsets from $\{c_1, ..., c_K\}$ as C_{src} and C_{tgt}, on which a single UMT model will be trained. The problem with single UMT model is obvious: each subset in $\{c_1, c_2, ..., c_K\}$ potentially represents a specific domain. The UMT model trained on the single C_{src} can thus only be able to properly paraphrase sentences from the C_{src} domain. To cover the full domain, we propose to train K UMT models, denoted by $\{M_1, M_2, ..., M_K\}$, where K is the number of clusters. Each of the trained UMT models uses a different $c \in \{c_1, c_2, ..., c_K\}$ as C_{src}, paired with a randomly selected C_{tgt}.

To paraphrase sentence s, we need to find its corresponding paraphrase generation model $M \in \{M_1, M_2, ..., M_K\}$, which takes s as the input and outputs its paraphrase. We first select the $C_{src} \in \{c_1, c_2, ..., c_K\}$ that s belongs to. Next, we pick that model M trained using C_{src} as sources, and use M to generate the output.

For LDA, C_{src} is the topic that generates s with the largest probability:

$$C_{src} = \underset{c \in \{c_1, c_2, ..., c_K\}}{\arg\max} \; p(s|c) \tag{7}$$

For the K-means model, C_{src} is the cluster whose center is closest to s:

$$C_{src} = \underset{c \in \{c_1, c_2, ..., c_K\}}{\arg\min} \; ||h_s - \mu_c||^2 \tag{8}$$

where μ_c denotes the center of the cluster c.

We follow Lample et al. (2017) to implement each of the UMT models. We used the transformer (Vaswani et al., 2017) as the backbone instead of LSTMs in Bahdanau et al. (2014), where the number of encoder blocks, decoder blocks, the number of heads, d_{model} and d_{ff} are respectively set to 6, 6, 8, 512 and 2,048. For UMT models based on specific C_{src} and C_{tgt}, both the encoder and the decoder are trained using Adam (Kingma and Ba, 2014), with the learning rate set to 0.00025, β_1 set to 0.5. We evenly alternate between the encoder-decoder and the discriminator.

Unifying Ms into a Surrogate Model We need to maintain K different domain-specific UMT models, which is both memory costly and computationally intensive, especially for online services. We thus propose to unify different Ms into a single surrogate one. For each sentence s in a selected corpus, we first find the cluster C_{src} it belongs to using LDA or K-means described above, and then we use the model M trained on C_{src} to generate the paraphrase of s. In this way, we are able to collect massive amounts of pseudo-labeled paraphrase pairs by treating the original sentence s as the source and the produced paraphrase as the target. We collected a total number of 25 million pairs. Human interventions can happen in this stage, where we can design filtering functions to remove pairs with undesired properties. Here, human interventions involve (1) removing pairs with identical source and target; (2) removing targets two times longer than sources. 16 million pairs remain after filtering.

We train a SEQ2SEQ model (Sutskever et al., 2014; Vaswani et al., 2017) (referred to as *UMT-Multi*) on the remaining pseudo-labeled data, which is used as the ultimate paraphrase generation model. We use the Transformer-base (Vaswani et al., 2017) as the model backbone, where the number of encoder blocks, decoder blocks, the number of heads, d_{model} and d_{ff} are respectively set to 6, 6, 8, 512 and 2,048. We use Adam (Kingma and Ba, 2014) with learning rate of 1e-4, β_1 = 0.9, β_2 = 0.999, and a warmup step of 4K. Batch size is set to 256. This model can be directly used in the unsupervised learning setup. An overview of deriving the *UMT-Multi* model is shown in Figure 1. Up to now, *UMT-Multi* is purely based on unlabeled commoncrawl corpus.

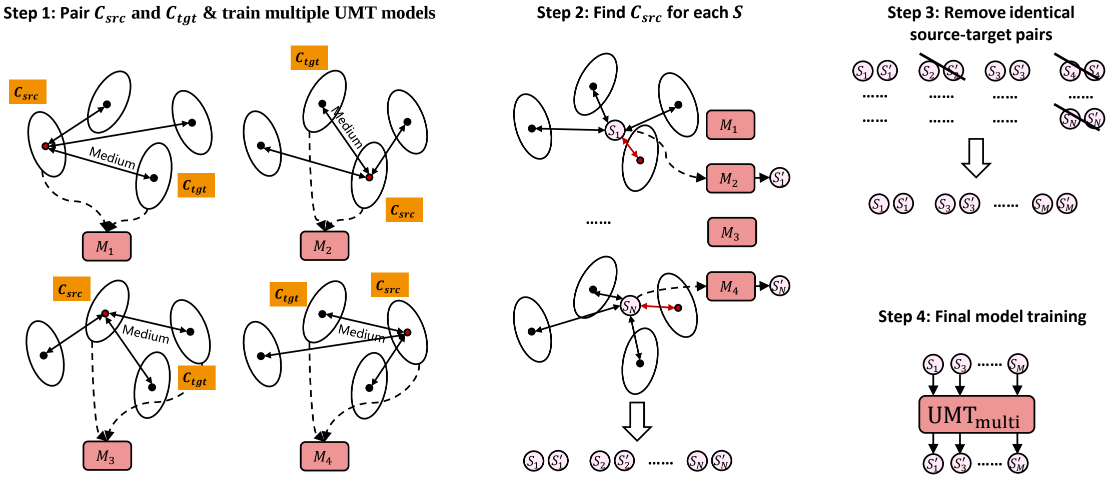
Step 1: Pair C_{src} and C_{tgt} & train multiple UMT models
Step 2: Find C_{src} for each S
Step 3: Remove identical source-target pairs
Step 4: Final model training
C_{src}
C_{tgt}
Medium
M_1
M_2
M_3
M_4
S_1
S_N
S_1'
S_N'
S_M
S_M'
UMT$_{\text{multi}}$

Figure 1: An overview of deriving the *UMT-Multi* model. Step 1: First, for each cluster c_i, we treat it as C_{src} and find its corresponding cluster C_{tgt}, and then train a UMT model on (C_{src}, C_{tgt}). The number of total UMT models is K (in this figure, $K = 4$). Step 2: For a given input sentence s, we first select the C_{src} that s belongs to, and use the model trained on C_{src} to generate the paraphrase of s. This process goes on over the entire corpus, leading to a pseudo labeled dataset of paraphrase pairs. Step 3: Human intervenes by removing paraphrase pairs whose inputs and output are the same, and outputs are two times longer than sources. Step 4: Training the single *UMT-Multi* model using the dataset after filtering. Step 5 (optional): fine-tuning the *UMT-Multi* model on the supervised paraphrase dataset in the supervised setup.

4.3 Supervised Setup

For the supervised setup, where we have pairs of paraphrases containing sources from a source domain and paraphrases of sources from a target domain, we can fine-tune the pretrained *UMT-Multi* model on the supervised paraphrase pairs, where we initialize the model using the *UMT-Multi* model, and run additional iterations on the supervised dataset. The fine-tuned model thus shares the structure with *UMT-Multi*. Again, we use Adam (Kingma and Ba, 2014) for fine-tuning, with $\beta_1 = 0.9$, $\beta_2 = 0.98$. Batch size, learning rate and the number of iterations are treated as hyper-parameters and tuned on the dev set. At test time, beam search (Sutskever et al., 2014; Li et al., 2016a) is used when decoding.

An additional use of the gold labeled paraphrase datasets is to help to select $C_{tgt} \in \{c_1, c_2, ..., c_K\}$ that best aligns with C_{src}, while in the unsupervised setup, we can only randomly pair C_{src} and C_{tgt} due to the lack of training signals for pairing. In the most straightforward setup, for each $C_{src} \in \{c_1, c_2, ..., c_K\}$, we can construct $K - 1$ pairs (C_{src}, c) by treating all $c \in \{c_1, c_2, ..., c_K\}, c \neq C_{src}$ as C_{tgt}. Next, we train $K - 1$ UMT models based on the pairs, and select the model that achieves the highest evaluation score on the labeled dataset. This strategy leads to a total number of $K \times (K - 1)$ models to be trained, which is computationally prohibitive. We propose a simplified learning model that maps the distance between C_{src} and C_{tgt} as inputs to output the evaluation score (here we use iBLEU) on the labeled dataset. Specifically, we randomly select L pairs, where $L \ll K \times (K - 1)$. We train L UMT models on the selected dataset pairs. Using the trained UMT models, we generate paraphrases for the labeled datasets, and obtain corresponding evaluation scores. Based on the distance between C_{src} and C_{tgt}, and the evaluation score $S(M_{(src,tgt)})$, we train a simple polynomial function F to learn to map the distance to the evaluation score:

$$S(M_{(src,tgt)}) = F(\text{Dis}(C_{src}, C_{tgt})) \qquad (9)$$

The function F can be then used to select C_{tgt} with highest predicted evaluation score for C_{src}.

5 Experiments

5.1 Experiment Setups

We consider both the supervised and unsupervised setups. There are two differences between the supervised and unsupervised setups: for the supervised setup, (1) the training data provides guidance on pairing C_{src} and C_{tgt}; and (2) the pretrained model will be used as initialization and later fine-tuned on the labeled dataset. Datasets that we use for evaluation include Quora, WikiAnswers (Fader et al., 2013), MSCOCO (Lin et al., 2014) and Twitter Liu et al. (2019a).

For the supervised setup, we compare our proposed model to the follow baselines: **ResidualLSTM** (Prakash et al., 2016b), **VAE-SVG-eq** (Gupta et al., 2018), **Pointer** (See et al., 2017), **Transformer** (Vaswani et al., 2017) and **DNPG** (Li et al., 2019). For the unsupervised setup, we use the following models for comparison: **VAE** (Bowman et al., 2016), **Lag VAE** (He et al., 2019), **CGMH** (Miao et al., 2019) and **UPSA** (Liu et al., 2019a). Results for VAE, Lag VAE, CGMH and UPSA on different datasets are copied from Miao et al. (2019) and Liu et al. (2019a). Results for ResidualLSTM, VAE-SVG-eq, Pointer, Transformer on various datasets are copied from Li et al. (2019). We leave details of these datasets, baselines and training in Appendix 7.

We are particularly interested in comparing the proposed model with bilingual MT based models. BT is trained end-to-end on WMT'14 En↔Fr.[2] A paraphrase pair is obtained by pairing the English sentence in the original dataset and the translation of the French sentence. Next we train a Transformer-large model on paraphrase pairs. The model is used as initialization to be further fine-tuned on the labeled dataset. We also use WMT-14 En-Zh for reference purposes. We use BLEU (Papineni et al., 2002), iBLEU (Sun and Zhou, 2012) and ROUGE scores (Lin, 2004) for evaluation.

5.2 Results

In-domain Results We first show the in-domain results in Table 1. We can observe that across all datasets and under both the supervised and unsupervised setups, the proposed UMT model significantly outperforms than baselines. As expected, multiple UMT models perform better than a single UMT model as the former is more flexible at selecting the correct domain C_{src} for an input sentence. We can also observe that the BT model is able to achieve competitive results, which shows that back-translation serves as a strong and simple baseline for paraphrase generation. The BT model trained on En-Fr consistently outperforms the one trained on En-Zh, and this is because that En-Zh translation is a harder task than En-Fr due to the greater grammars difference between the two languages.

	Model	iBLEU	BLEU	R1	R2
Supervised	*Quora*				
	ResidualLSTM	12.67	17.57	59.22	32.40
	VAE-SVG-eq	15.17	20.04	59.98	33.30
	Pointer	16.79	22.65	61.96	36.07
	Transformer	16.25	21.73	60.25	33.45
	DNPG	18.01	25.03	63.73	37.75
	BT(En-Fr)	18.04	25.34	63.82	37.92
	BT(En-Zh)	17.67	24.90	63.32	37.38
	UMT-Single	17.70	24.97	63.65	37.77
	UMT-Multi	**18.78**	**26.49**	**64.12**	**38.31**
	Wikianswers				
	ResidualLSTM	22.94	27.36	48.52	18.71
	VAE-SVG-eq	26.35	32.98	50.93	19.11
	Pointer	31.98	39.36	57.19	25.38
	Transformer	27.70	33.01	51.85	20.70
	DNPG	34.15	41.64	57.32	25.88
	BT(En-Fr)	34.55	41.90	57.84	26.44
	BT(En-Zh)	33.98	41.04	56.37	25.60
	UMT-Single	34.50	41.72	57.58	26.31
	UMT-Multi	**36.04**	**42.94**	**58.71**	**27.35**
Unsupervised	*Quora*				
	VAE	8.16	13.96	44.55	22.64
	Lag VAE	8.73	15.52	49.20	26.07
	CGMH	9.94	15.73	48.73	26.12
	UPSA	12.03	18.21	59.51	32.63
	BT(En-Fr)	11.98	17.84	59.03	32.11
	BT(En-Zh)	11.33	17.02	56.19	31.08
	UMT-Single	11.47	17.21	56.35	31.27
	UMT-Multi	**13.10**	**18.98**	**59.90**	**33.04**
	Wikianswers				
	VAE	17.92	24.13	31.87	12.08
	Lag VAE	18.38	25.08	35.65	13.21
	CGMH	20.05	26.45	43.31	16.53
	UPSA	24.84	32.39	54.12	21.45
	BT(En-Fr)	23.55	31.10	52.03	20.86
	BT(En-Zh)	22.60	30.12	51.29	20.11
	UMT-Single	23.01	30.62	51.79	20.35
	UMT-Multi	**25.90**	**33.80**	**54.52**	**23.48**
	MSCOCO				
	VAE	7.48	11.09	31.78	8.66
	Lag VAE	7.69	11.63	32.20	8.71
	CGMH	7.84	11.45	32.19	8.67
	UPSA	9.26	14.16	37.18	11.21
	BT(En-Fr)	8.15	13.78	36.30	10.48
	BT(En-Zh)	7.80	11.97	32.40	9.21
	UMT-Single	8.21	13.99	36.52	10.75
	UMT-Multi	**9.70**	**15.42**	**38.51**	**12.39**
	Twitter				
	VAE	2.92	3.46	15.13	3.40
	Lag VAE	3.15	3.74	17.20	3.79
	CGMH	4.18	5.32	19.96	5.44
	UPSA	4.93	6.87	28.34	8.53
	BT(En-Fr)	4.32	5.97	26.37	7.59
	BT(En-Zh)	4.15	5.40	25.83	7.32
	UMT-Single	4.40	6.11	26.89	7.78
	UMT-Multi	**5.35**	**7.80**	**29.74**	**9.88**

Table 1: In-domain performances of different models for both supervised and unsupervised setups.

Domain-adapted Results We test the model's domain adaptation ability on Quora and Wikianswers. Table 2 shows the results. We can see that UMT-multi performs significantly better than baselines, including UMT-single, showing the better

[2] Wieting et al. (2017); Wieting and Gimpel (2017) suggested little difference among Czech, German, and French as source languages for back-translation. We use En↔Fr since it contains more parallel data than other language pairs.

Model	iBLEU	BLEU	R1	R2
	Wikianswers→Quora			
Pointer	5.04	6.96	41.89	12.77
Transformer+Copy	6.17	8.15	44.89	14.79
DNPG	10.39	16.98	56.01	28.61
BT(En-Fr)	12.14	17.98	59.42	32.44
BT(En-Zh)	11.43	17.21	56.65	31.45
UMT-Single	11.86	17.49	57.01	32.44
UMT-Multi	**13.62**	**19.48**	**61.04**	**33.85**
	Quora→Wikianswers			
Pointer	21.87	27.94	53.99	20.85
Transformer+Copy	23.25	29.22	53.33	21.02
DNPG	25.60	35.12	56.17	23.65
BT(En-Fr)	25.77	35.30	56.41	23.78
BT(En-Zh)	24.84	34.19	55.71	22.60
UMT-Single	25.43	34.70	56.10	23.31
UMT-Multi	**26.85**	**36.64**	**57.45**	**24.60**

Table 2: Domain-adapted performances of different models. "R1" stands for ROUGE-1 and "R2" stands for ROUGE-2.

ability of UMT-multi for domain adaptation.

5.3 Human Evaluation

To further validate the performance of the proposed model, we randomly sample 500 sentences from Quora test set for human evaluation. The input sentence and its two paraphrases respectively generated by the UMT model and the BT model (En-Fr) are assigned to two human annotators at Amazon Mechanical Turk (AMT), with "> 95% HIT approval rate". Annotators are asked to judge which output is better in terms of three aspects: (1) semantics: whether the two sentences are the same semantic meaning; (2) diversity: whether the two sentences are diverse in expressions; and (3) fluency: whether the generated paraphrase is fluent. Ties are allowed. If the two annotators' evaluations do not agree with each other, the job will be assigned to one more annotator, and we take the majority as the final result.[3] Comparing with BT, the proportions of win, tie and lose for the proposed UMT-model are respectively 41%, 36%, and 22%, demonstrating its superiority over BT models.

5.4 Examples

Table 3 presents sampled paraphrases from the BT and UMT models. From these examples, we can identify several intrinsic drawbacks of the BT model that the UMT model can circumvent: (1) for the first example, the tense from the BT paraphrase model based on En-Zh translation is incorrect. This is because the Chinese language expresses tense in a more implicit way. This leads the model to make mistake in tense when Chinese is translated back to English. The UMT model does not have this issue; (2) for the second example, BT model directly copies the input, this is because the En-Fr can perfectly map the meaning in two languages with no expression variations. Due to the blackbox nature of MT models, it is hard to intervene with the process to avoid producing the same copy. Instead, for the proposed UMT framework, developers can intervene with the model in both clustering stage and data filtering stage. (3) For the third example, the BT model changes the meaning of the original sentence, which is due to the mistake made by the translation model. These mistakes are sometimes inevitable due to the limitation of current MT models, but can be fixed in the proposed system.

6 Ablation Study

In this section, we perform comprehensive ablation studies on Wikianswers dataset for understanding behaviors of the proposed model. And we report iBLEU score for comparison.

Size of C for UMT Training First, we explore how the size of C, the CommonCrawl corpus used for dataset split and UMT training, affects downstream performances. Table 4 shows the results, where the size is respectively 10M, 100M, 1B and 10B. We can observe that with more training data, the performance significantly improves. This is because the trained model can better learn to align sentences between different clusters.

The Number of LDA Topics Table 5 presents the influence of the number of LDA clusters. The trend is clear: more topics lead to better performances. This is because the model with more topics has a stronger ability of disentangling very similar sentences in the original corpus C, and thus avoids copying. It is worth noting that more topics means training more UMT models before unifying them, leading to greater computational intensity.

Pairing C_{src} and C_{tgt} In our main experiments, we randomly select C_{tgt} given C_{src}. It would be interesting to see the effects of different cluster selection strategies. We consider four strategies: *Largest* (select C_{tgt} with the largest distance to C_{src}), *Medium* (select C_{tgt} with the medium distance to C_{src}), *Smallest* (select C_{tgt} with the smallest distance to C_{src}) and for referring purposes,

[3] If the three annotators all disagree, we discard the instance.

Input	**BT**	**UMT**
Time is the most accurate test of how puberty is going to progress.	Time is the most accurate test of how puberty is going to progress. (En-Fr)	How adolescence develops is most accurately test by time.
GM told truck owners about the defect in the first half of October.	GM owners told truck about the defect in the first half of October . (En-Fr)	GM informed truck owners of the defect in the first half of October .
To keep him alive , well , there 's no reason to keep him alive .	To keep him alive , well , there 's no reason to keep him alive . (En-Fr)	To let him live , well , there 's no reason to let him live .
Washington did not have the formal education his elder brothers received at Appleby Grammar School in England, but he did learn mathematics, trigonometry, and land surveying .	Washington did not pursue the same formal education as his older brothers at England's Appleby Grammar School, but he did study geometry, trigonometry, and land surveying .	Unlike his older brothers, who studied at England's Appleby Grammar School, Washington did not receive formal education but studied mathematics, trigonometry, and land surveying .

Table 3: Sampled paraphrases from the BT and UMT models.

Size	**10M**	**100M**	**1B**	**10B**
Unsupervised. UMT-Multi	15.5	21.1	24.2	**25.9**

Table 4: The effect of data size of C for training UMT.

# LDA Topic	**5**	**20**	**50**	**80**
Unsupervised. UMT-Multi	14.9	22.4	24.9	**25.9**

Table 5: The effect of number of LDA topics.

Strategy	**Unsuper. UMT-Multi**	**Super. UMT-Multi**
Random	25.9	35.4
Largest	24.7	35.1
Medium	25.8	35.7
Smallest	25.3	35.5
Supervised	**26.3**	**36.0**

Table 6: The effect of different strategies to pair C_{src} and C_{tgt}.

	LDA		**K-means**	
Clustering	**Single**	**Multi**	**Single**	**Multi**
Uns.	23.0	**25.9**	21.9	24.2
Su.	34.5	**36.0**	32.1	34.2

Table 7: The effect of different clustering methods for C. "Uns." means we use the unsupervised setup and "Su." represents the supervised setup.

Supervised (select C_{tgt} using the supervised strategy proposed). In the *real* unsupervised setup, the supervised strategy cannot be readily applied since we have no access to supervised labels. We list performance for *supervised* here for referring purpose.

Table 6 shows the results. For both supervised and unsupervised setups, *Supervised* performs the best against the other strategies, especially under the unsupervised setup. The difference in performances between these strategies is greater for the unsupervised setup than the supervised setup. This is because supervised training serves to compensate the performance gap due to the presence of labeled training data. We find that the random strategy outperforms both *Largest* and *Smallest*. For *Largest* , this is because *Largest* leads to very different paired clusters, having the risk that some sentences in C_{src} might not have correspondences in C_{tgt}. For *Smallest*, since paired clusters are pretty close, sentences in C_{src} are more likely to have copies in C_{tgt}. *Largest* and *Smallest* leads to inferior performances. *random* performs comparable to *medium*.

Clustering Methods We study the effect of different clustering methods, i.e., LDA and K-means. Table 7 shows the results. As can be seen, for both supervised and unsupervised setups, the model trained with LDA consistently performs better than the model trained with K-means. We think there are potentially two reasons: (1) the BERT representations, on which clustering relies, cannot well represent sentence semantics for clustering (Reimers and Gurevych, 2019); and (2) the K-means model for sentence clustering operates at a relatively low level of semantics (i.e., sentence level), while LDA takes into the more global document level information. Due to the entanglement of sentence semantics in C, it is hard for K-means to separate sentences apart, or if it can, it takes long until convergence.

7 Conclusion

In this paper, we propose a new framework for paraphrase generation by treating the task as unsupervised machine translation (UMT). The proposed framework first splits a large unlabeled corpus into multiple sub-datasets and then trains one or multiple UMT models based on one or more pairs of these sub-datasets. Experiments and ablation studies under supervised and unsupervised setups demonstrate the effectiveness of the proposed framework.

Acknowledgement

We would like to thank anonymous reviewers for their comments and suggestions. This work is supported by the Key R & D Projects of the Ministry of Science and Technology (2020YFC0832500).

References

Mikel Artetxe, Gorka Labaka, and Eneko Agirre. 2018. Unsupervised statistical machine translation. *arXiv preprint arXiv:1809.01272*.

Mikel Artetxe, Gorka Labaka, and Eneko Agirre. 2019. An effective approach to unsupervised machine translation. *arXiv preprint arXiv:1902.01313*.

Mikel Artetxe, Gorka Labaka, Eneko Agirre, and Kyunghyun Cho. 2017. Unsupervised neural machine translation. *arXiv preprint arXiv:1710.11041*.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate.

Colin Bannard and Chris Callison-Burch. 2005. Paraphrasing with bilingual parallel corpora. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL'05)*, pages 597–604.

Regina Barzilay. 2004. Information fusion for multidocument summarization: Paraphrasing and generation.

Jonathan Berant and Percy Liang. 2014. Semantic parsing via paraphrasing. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1415–1425.

David M Blei, Andrew Y Ng, and Michael I Jordan. 2003. Latent dirichlet allocation. *the Journal of machine Learning research*, 3:993–1022.

Samuel R. Bowman, Luke Vilnis, Oriol Vinyals, Andrew Dai, Rafal Jozefowicz, and Samy Bengio. 2016. Generating sentences from a continuous space. In *Proceedings of The 20th SIGNLL Conference on Computational Natural Language Learning*, pages 10–21, Berlin, Germany. Association for Computational Linguistics.

Chris Callison-Burch, Philipp Koehn, and Miles Osborne. 2006. Improved statistical machine translation using paraphrases. In *Proceedings of the Human Language Technology Conference of the NAACL, Main Conference*, pages 17–24.

Ziqiang Cao, Chuwei Luo, Wenjie Li, and Sujian Li. 2016. Joint copying and restricted generation for paraphrase.

Mingda Chen, Qingming Tang, Sam Wiseman, and Kevin Gimpel. 2019. Controllable paraphrase generation with a syntactic exemplar. *arXiv preprint arXiv:1906.00565*.

Kevin Clark, Minh-Thang Luong, Quoc V Le, and Christopher D Manning. 2020. Electra: Pre-training text encoders as discriminators rather than generators. *arXiv preprint arXiv:2003.10555*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Li Dong, Jonathan Mallinson, Siva Reddy, and Mirella Lapata. 2017. Learning to paraphrase for question answering. *arXiv preprint arXiv:1708.06022*.

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, Sofia, Bulgaria. Association for Computational Linguistics.

Yao Fu, Yansong Feng, and John P Cunningham. 2019. Paraphrase generation with latent bag of words. In *Advances in Neural Information Processing Systems*, pages 13645–13656.

Zhenxin Fu, Xiaoye Tan, Nanyun Peng, Dongyan Zhao, and Rui Yan. 2018. Style transfer in text: Exploration and evaluation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32.

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.

Xavier Garcia, Pierre Foret, Thibault Sellam, and Ankur P Parikh. 2020. A multilingual view of unsupervised machine translation. *arXiv preprint arXiv:2002.02955*.

Ian J Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. 2014. Generative adversarial networks. *arXiv preprint arXiv:1406.2661*.

Tanya Goyal and Greg Durrett. 2020. Neural syntactic preordering for controlled paraphrase generation. *arXiv preprint arXiv:2005.02013*.

Ankush Gupta, Arvind Agarwal, Prawaan Singh, and Piyush Rai. 2018. A deep generative framework for paraphrase generation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32.

Junxian He, Daniel Spokoyny, Graham Neubig, and Taylor Berg-Kirkpatrick. 2019. Lagging inference networks and posterior collapse in variational autoencoders. *arXiv preprint arXiv:1901.05534*.

Chaitra Hegde and Shrikumar Patil. 2020. Unsupervised paraphrase generation using pre-trained language models. *arXiv preprint arXiv:2006.05477*.

Ari Holtzman, Jan Buys, M. Forbes, and Yejin Choi. 2020. The curious case of neural text degeneration. *ArXiv*, abs/1904.09751.

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. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 6521–6528.

Zhiting Hu, Zichao Yang, Xiaodan Liang, Ruslan Salakhutdinov, and Eric P Xing. 2017. Toward controlled generation of text. In *International Conference on Machine Learning*, pages 1587–1596. PMLR.

Mohit Iyyer, John Wieting, Kevin Gimpel, and Luke Zettlemoyer. 2018. Adversarial example generation with syntactically controlled paraphrase networks. *arXiv preprint arXiv:1804.06059*.

Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

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

Guillaume Lample and Alexis Conneau. 2019. Cross-lingual language model pretraining. *arXiv preprint arXiv:1901.07291*.

Guillaume Lample, Alexis Conneau, Ludovic Denoyer, and Marc'Aurelio Ranzato. 2017. Unsupervised machine translation using monolingual corpora only. *arXiv preprint arXiv:1711.00043*.

Guillaume Lample, Myle Ott, Alexis Conneau, Ludovic Denoyer, and Marc'Aurelio Ranzato. 2018. Phrase-based & neural unsupervised machine translation. *arXiv preprint arXiv:1804.07755*.

Wuwei Lan, Siyu Qiu, Hua He, and Wei Xu. 2017. A continuously growing dataset of sentential paraphrases. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1224–1234, Copenhagen, Denmark. Association for Computational Linguistics.

Jiwei Li, Will Monroe, and Dan Jurafsky. 2016a. A simple, fast diverse decoding algorithm for neural generation. *arXiv preprint arXiv:1611.08562*.

Jiwei Li, Will Monroe, Alan Ritter, Michel Galley, Jianfeng Gao, and Dan Jurafsky. 2016b. Deep reinforcement learning for dialogue generation. *arXiv preprint arXiv:1606.01541*.

Juncen Li, Robin Jia, He He, and Percy Liang. 2018a. Delete, retrieve, generate: A simple approach to sentiment and style transfer. *arXiv preprint arXiv:1804.06437*.

Zichao Li, Xin Jiang, Lifeng Shang, and Hang Li. 2018b. Paraphrase generation with deep reinforcement learning. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3865–3878, Brussels, Belgium. Association for Computational Linguistics.

Zichao Li, Xin Jiang, Lifeng Shang, and Qun Liu. 2019. Decomposable neural paraphrase generation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3403–3414, Florence, Italy. Association for Computational Linguistics.

Zuchao Li, Hai Zhao, Rui Wang, Masao Utiyama, and Eiichiro Sumita. 2020. Reference language based unsupervised neural machine translation. *arXiv preprint arXiv:2004.02127*.

Chin-Yew Lin. 2004. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain. Association for Computational Linguistics.

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. 2019a. Unsupervised paraphrasing by simulated annealing. *arXiv preprint arXiv:1909.03588*.

Yinhan Liu, Jiatao Gu, Naman Goyal, Xian Li, Sergey Edunov, Marjan Ghazvininejad, Mike Lewis, and Luke Zettlemoyer. 2020. Multilingual denoising pre-training for neural machine translation. *arXiv preprint arXiv:2001.08210*.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019b. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Minh-Thang Luong, Ilya Sutskever, Quoc V. Le, Oriol Vinyals, and Wojciech Zaremba. 2015. Addressing the rare word problem in neural machine translation.

Shuming Ma, Xu Sun, Wei Li, Sujian Li, Wenjie Li, and Xuancheng Ren. 2018. Query and output: Generating words by querying distributed word representations for paraphrase generation. In *Proceedings of* *the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 196–206, New Orleans, Louisiana. Association for Computational Linguistics.

Jonathan Mallinson, Rico Sennrich, and Mirella Lapata. 2017. Paraphrasing revisited with neural machine translation. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, pages 881–893.

Benjamin Marie and Atsushi Fujita. 2018. Unsupervised neural machine translation initialized by unsupervised statistical machine translation. *arXiv preprint arXiv:1810.12703*.

Yuxian Meng, Xiang Ao, Qing He, Xiaofei Sun, Qinghong Han, Fei Wu, Jiwei Li, et al. 2021. Conrpg: Paraphrase generation using contexts as regularizer. *arXiv preprint arXiv:2109.00363*.

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.

Jonas Mueller, David Gifford, and Tommi Jaakkola. 2017. Sequence to better sequence: continuous revision of combinatorial structures. In *International Conference on Machine Learning*, pages 2536–2544. PMLR.

Tong Niu, Semih Yavuz, Yingbo Zhou, Huan Wang, Nitish Shirish Keskar, and Caiming Xiong. 2020. Unsupervised paraphrase generation via dynamic blocking. *arXiv preprint arXiv:2010.12885*.

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.

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

Aaditya Prakash, Sadid A. Hasan, Kathy Lee, Vivek Datla, Ashequl Qadir, Joey Liu, and Oladimeji Farri. 2016b. Neural paraphrase generation with stacked residual LSTM networks. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 2923–2934, Osaka, Japan. The COLING 2016 Organizing Committee.

Lihua Qian, Lin Qiu, Weinan Zhang, Xin Jiang, and Yong Yu. 2019. Exploring diverse expressions for paraphrase generation. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3164–3173.

Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. 2018. Improving language understanding by generative pre-training.

Marc'Aurelio Ranzato, Sumit Chopra, Michael Auli, and Wojciech Zaremba. 2015. Sequence level training with recurrent neural networks. *arXiv preprint arXiv:1511.06732*.

Sujith Ravi and Kevin Knight. 2011. Deciphering foreign language. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 12–21, Portland, Oregon, USA. Association for Computational Linguistics.

Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. *arXiv preprint arXiv:1908.10084*.

Shuo Ren, Yu Wu, Shujie Liu, Ming Zhou, and Shuai Ma. 2020. A retrieve-and-rewrite initialization method for unsupervised machine translation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 3498–3504, Online. Association for Computational Linguistics.

Shuo Ren, Zhirui Zhang, Shujie Liu, Ming Zhou, and Shuai Ma. 2019. Unsupervised neural machine translation with smt as posterior regularization. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 241–248.

Annette Rios Gonzales, Laura Mascarell, and Rico Sennrich. 2017. Improving word sense disambiguation in neural machine translation with sense embeddings. In *Proceedings of the Second Conference on Machine Translation*, pages 11–19, Copenhagen, Denmark. Association for Computational Linguistics.

Aurko Roy and David Grangier. 2019. Unsupervised paraphrasing without translation. *arXiv preprint arXiv:1905.12752*.

Abigail See, Peter J. Liu, and Christopher D. Manning. 2017. Get to the point: Summarization with pointer-generator networks. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1073–1083, Vancouver, Canada. Association for Computational Linguistics.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Improving neural machine translation models with monolingual data. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 86–96, Berlin, Germany. Association for Computational Linguistics.

Tianxiao Shen, Tao Lei, Regina Barzilay, and Tommi Jaakkola. 2017. Style transfer from non-parallel text by cross-alignment.

A. B. Siddique, Samet Oymak, and Vagelis Hristidis. 2020. Unsupervised paraphrasing via deep reinforcement learning. *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*.

Alex Sokolov and Denis Filimonov. 2020. Neural machine translation for paraphrase generation. *arXiv preprint arXiv:2006.14223*.

Kaitao Song, Xu Tan, Tao Qin, Jianfeng Lu, and Tie-Yan Liu. 2019. Mass: Masked sequence to sequence pre-training for language generation. *arXiv preprint arXiv:1905.02450*.

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 (Volume 2: Short Papers)*, pages 38–42.

Xiaofei Sun, Yuxian Meng, Xiang Ao, Fei Wu, Tianwei Zhang, Jiwei Li, and Chun Fan. 2021a. Sentence similarity based on contexts. *arXiv preprint arXiv:2105.07623*.

Zijun Sun, Xiaoya Li, Xiaofei Sun, Yuxian Meng, Xiang Ao, Qing He, Fei Wu, and Jiwei Li. 2021b. Chinesebert: Chinese pretraining enhanced by glyph and pinyin information. *arXiv preprint arXiv:2106.16038*.

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

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

Su Wang, Rahul Gupta, Nancy Chang, and Jason Baldridge. 2018. A task in a suit and a tie: paraphrase generation with semantic augmentation.

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*.

John Wieting, Jonathan Mallinson, and Kevin Gimpel. 2017. Learning paraphrastic sentence embeddings from back-translated bitext. *arXiv preprint arXiv:1706.01847*.

Sam Witteveen and Martin Andrews. 2019. Paraphrasing with large language models. *arXiv preprint arXiv:1911.09661*.

Jiawei Wu, Xin Wang, and William Yang Wang. 2019. Extract and edit: An alternative to back-translation for unsupervised neural machine translation. *arXiv preprint arXiv:1904.02331*.

Qiongkai Xu, Juyan Zhang, Lizhen Qu, Lexing Xie, and Richard Nock. 2018. D-page: Diverse paraphrase generation. *arXiv preprint arXiv:1808.04364*.

Qian Yang, Dinghan Shen, Yong Cheng, Wenlin Wang, Guoyin Wang, Lawrence Carin, et al. 2019. An end-to-end generative architecture for paraphrase generation. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3123–3133.

Zhen Yang, Wei Chen, Feng Wang, and Bo Xu. 2018. Unsupervised neural machine translation with weight sharing. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 46–55, Melbourne, Australia. Association for Computational Linguistics.

Jinhua Zhu, Yingce Xia, Lijun Wu, Di He, Tao Qin, Wengang Zhou, Houqiang Li, and Tie-Yan Liu. 2020. Incorporating bert into neural machine translation. *arXiv preprint arXiv:2002.06823*.

A Datasets

(1) **Quora**[4]: The Quora dataset contains 140K parallel paraphrases of questions and 260K non-parallel sentences collected from the question answering website Quora[5]. We follow the standard setup in Miao et al. (2019) and use 3K/30K paraphrase pairs respectively for validation and test.
(2) **Wikianswers**: The WikiAnswers corpus (Fader et al., 2013) contains clusters of questions tagged by WikiAnswers users as paraphrases. It contains a total number of 2.3M paraphrase pairs. We follow Liu et al. (2019a) to randomly pick 5K pairs for validation and 20K for test.[6]
(3) **MSCOCO**: The MSCOCO dataset (Lin et al., 2014) contains over 500K paraphrase pairs for 120K image captions, with each image caption annotated by five annotators. We follow the dataset split and the evaluation protocol in Prakash et al. (2016b), where only image captions with fewer than 15 words are considered.
(4) **Twitter**: The Twitter dataset is collected via linked tweets through shared URLs (Lan et al., 2017), which originally contains 50K paraphrase pairs. We follow the data split in Liu et al. (2019a), where 10% of the training data is used as validation and the test set only contains sentence pairs that are labeled as "paraphrases".

B Baselines

For the supervised setup, we compare our proposed model to the follow baselines:
(1) **ResidualLSTM**: Prakash et al. (2016b) deepened the LSTM network by stacking multiple layers with residual connection. This deep SEQ2SEQ model is trained on labeled paraphrase datasets.
(2) **VAE-SVG-eq**: Gupta et al. (2018) combined VAEs with LSTMs to generate paraphrases in a SEQ2SEQ generative style.
(3) **Pointer**: See et al. (2017) augmented the standard SEQ2SEQ model by using a pointer, i.e., the copy mechanism. Word in the input sentence can be directly copied as the current decoded word.
(4) **Transformer**: Vaswani et al. (2017) proposed the Transformer architecture which is based on the self-attention mechanism.
(5) **DNPG**: Li et al. (2019) proposed a Transformer-based model that learns and generates paraphrases at different levels of granularity, i.e., from the lexical to phrasal and then to sentential levels.

For the unsupervised setup, we use the following models for comparison:
(1) **VAE**: Bowman et al. (2016) proposed variational auto-encoders (VAEs) to generate sentences from a continuous space. By minimizing the reconstruction loss between the input sentence and the output sentence, VAEs are able to sample paraphrases from the continuous space.
(2) **Lag VAE**: To overcome the posterior collapse issue of VAEs, He et al. (2019) proposed to aggressively optimize the inference network by performing multiple updates before reverting back to basic VAE training.
(3) **CGMH**: Miao et al. (2019) used Metropolis–Hastings sampling to generate paraphrases, where a word can be deleted, replaced or inserted into the current sentence based on the sampling distribution.
(4) **UPSA**: Liu et al. (2019a) proposed to use simulated annealing to optimize the paraphrase generation model. The training objective is composed of three parts: semantic similarity, expression diversity and language fluency.

[4] https://www.kaggle.com/c/quora-question-pairs
[5] https://www.quora.com/
[6] Note that the selected data is different from Liu et al. (2019a) but is comparable in the statistical sense.