ConRPG: Paraphrase Generation using Contexts as Regularizer

Yuxian Meng[♣], Xiang Ao[▲], Qing He[▲], Xiaofei Sun[♣]
Qinghong Han[♣], Fei Wu[♦], Chun fan[♠★▼] and Jiwei Li[♦♣]
[♦]Zhejiang University
[♠]Computer Center of Peking University, [★]Peng Cheng Laboratory
[▼]National Biomedical Imaging Center, Peking University
[▲] Key Lab of Intelligent Information Processing of Chinese Academy of Sciences
[♣]Shannon.AI
{yuxian_meng, xiaofei_sun, qinghong_han, jiwei_li}@shannonai.com
{aoxiang, heqing}@ict.ac.cn, fanchun@pku.edu.cn, wufei@zju.edu.cn

Abstract

A long-standing issue with paraphrase generation is how to obtain reliable supervision signals. In this paper, we propose an unsupervised paradigm for paraphrase generation based on the assumption that the probabilities of generating two sentences with the same meaning given the same context should be the same. Inspired by this fundamental idea, we propose a pipelined system which consists of paraphrase candidate generation based on contextual language models, candidate filtering using scoring functions, and paraphrase model training based on the selected candidates.

The proposed paradigm offers merits over existing paraphrase generation methods: (1) using the context regularizer on meanings, the model is able to generate massive amounts of high-quality paraphrase pairs; and (2) using human-interpretable scoring functions to select paraphrase pairs from candidates, the proposed framework provides a channel for developers to intervene with the data generation process, leading to a more controllable model. Experimental results across different tasks and datasets demonstrate that the effectiveness of the proposed model in both supervised and unsupervised setups.

1 Introduction

Paraphrase generation (Prakash et al., 2016a; Cao et al., 2016; Ma et al., 2018; Wang et al., 2018) is the task of generating an output sentence which is semantically identical to a given input sentence but with variations in lexicon or syntax. It is a long-standing problem in the field of natural language processing (NLP) (McKeown, 1979; Meteer and Shaked, 1988; Quirk et al., 2004; Bannard and Callison-Burch, 2005a; Chen and Dolan, 2011) and has fundamental applications on end tasks such as semantic parsing (Berant and Liang, 2014), language model pretraining (Lewis et al., 2020) and question answering (Dong et al., 2017).

A long-standing challenge with paraphrase generation is to obtain reliable supervision signals. One way to resolve this issue is to manually annotate paraphrase pairs, which is both labor-intensive and expensive. Existing labeled paraphrase datasets (Lin et al., 2014; Fader et al., 2013; Lan et al., 2017) are either of small sizes or restricted in narrow domains. For example, the Quora dataset[1] contains 140K paraphrase pairs, the size of which is insufficient to build a large neural model. As another example, paraphrases in the larger MSCOCO (Lin et al., 2014) dataset are originally collected as image captions for object recognition, and repurposed for paraphrase generation. The domain for the MSCOCO dataset is thus restricted to captions depicting visual scenes.

Unsupervised methods, such as reinforcement learning (Li et al., 2018; Siddique et al., 2020) and auto-encoders (Bowman et al., 2016; Roy and Grangier, 2019), on the other hand, have exhibited their ability for paraphrase generation in the absence of annotated datasets. The core problem with existing unsupervised methods for paraphrase is the lack of an objective (or reward function in RL) that reliably measures the semantic relatedness between two diverse expressions in an unsupervised manner, with which the model can be trained to promote pairs with the same meaning but diverse expressions. For example, Hegde and Patil (2020) crafted unsupervised pseudo training examples by corrupting a sentence and then fed the corrupted one to a pretrained model as the input with the original sentence as the output. Since the model is restricted to learning to *reconstruct* corrupted

[1] https://www.kaggle.com/c/quora-question-pairs

Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, pages 2551–2562
November 7–11, 2021.

sentences, the generated paraphrases tend to be highly similar to the input sentences in terms of both wording and word orders. The issue in Hegde and Patil (2020) can be viewed as a microcosm of problems in existing unsupervised methods for paraphrase: we wish sentences to be diverse in expressions, but do not have a reliable measurement to avoid meaning change when expressions change. Additionally, the action of sentence corrupting can be less controllable.

In this work, we propose to address this issue by a new paradigm based on the assumption that the probabilities of generating two sentences with the same meaning based on the same context should be the same. With this core idea in mind, we propose a pipelined system which consists of the following steps: (1) paraphrase candidate generation by decoding sentences given its context using a language generation model; (2) candidate filtering based on scoring functions; and (3) paraphrase model training by training a SEQ2SEQ paraphrase generation model, which can be latter used for supervised finetuning on labeled datasets or directly used for unsupervised paraphrase generation.

The proposed paradigm offers the following merits over existing methods: (1) using the context regularizer on meanings, the model is able to generate massive amounts of high-quality paraphrase pairs; and (2) using human-interpretable ranking scores to select paraphrase pairs from candidates, the proposed framework provides a channel for developers to intervene with the data generation process, leading to a more controllable paraphrase model. Extensive experiments across different datasets under both supervised and unsupervised setups demonstrate the effectiveness of the proposed model.

2 Related Work

Supervised Methods for paraphrase generation rely on annotated paraphrase pairs to train the model. Iyyer et al. (2018); Li et al. (2019); Chen et al. (2019); Goyal and Durrett (2020) leveraged syntactic structures to generate diverse paraphrases with different syntax. Xu et al. (2018); Qian et al. (2019) used different semantic embeddings or generators to produce more diverse paraphrases. Kazemnejad et al. (2020) proposed a retrieval-based approach to retrieve paraphrase from a large corpus. Mallinson et al. (2017); Sokolov and Filimonov (2020) casted paraphrase generation as the task of machine translation. Mallinson et al. (2017); Wieting et al. (2017) extended the idea of bilingual pivoting for paraphrase generation where the input sentence is first translated into a foreign language, and then translated back as the paraphrase. Sokolov and Filimonov (2020) trained a MT model using multilingual parallel data and then finetuned the model using parallel paraphrase data.

Unsupervised Methods Li et al. (2018); Siddique et al. (2020) proposed to generate paraphrases using reinforcement learning, where certain rewarding criteria such as BLEU and ROUGE are optimized. Bowman et al. (2016); Yang et al. (2019) used the generative framework for paraphrase generation by training a variational auto-encoder (VAE) (Kingma and Welling, 2013) to optimize the lower bound of the reconstruction likelihood for an input sentence. Sentences sampled through the VAE's decoder can be regarded as paraphrases for an input sentence due to the reconstruction optimization target. Fu et al. (2019) similarly adopted a generative method but worked at the bag-of-words level. Other works explored paraphrase generation in an unsupervised manner by using vector quantised VAE (VQ-VAE) (Roy and Grangier, 2019), simulated annealing (Liu et al., 2019) or disentangled syntactic and semantic spaces (Bao et al., 2019). More recently, large-scale language model pretraining has also been proven to benefit paraphrase generation in both supervised learning (Witteveen and Andrews, 2019) and unsupervised learning (Hegde and Patil, 2020). Krishna et al. (2020) proposed diverse paraphrasing by warping the input's meaning through attribute transfer.

Regarding soliciting large-scale paraphrase datasets, Bannard and Callison-Burch (2005b) 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, inference sampling, and clustering. A relevant work to ours is Sun et al. (2021), which harnesses context to obtain sentence similarity. Sun et al. (2021) focuses on sentence

similarity rather than paraphrase generation.

3 Model

The key point of the proposed paradigm is to generate paraphrases based on the same context. This can be done in the following pipelined system: (1) we first train a contextual language generation model (*context-LM*) that predicts sentences given left and right contexts; (2) the pretrained contextual generation model decodes multiple sentences given the same context, and decoded sentences are treated as paraphrase candidates; (3) due to the fact that decoded sentences can be extremely noisy, further filtering is needed; (4) given the selected paraphrase, a SEQ2SEQ model (Sutskever et al., 2014) is trained using one sentence of the paraphrase pair as the source and the other as the target; the SEQ2SEQ model can be directly taken for the use of paraphrase in the unsupervised learning setup, or used as initialization to be further finetuned on labeled paraphrase datasets in the supervised learning setup. An overview of the proposed framework in depicted in Figure 1, the constituent unit of which will be detailed in order below.

3.1 Training context-LM

Let $\boldsymbol{c}_i = \{w_{i,1}, w_{i,2}, \cdots, w_{i,n}\}$ denote the i-th sentence within the given text, where n is number of words in c_j. $\boldsymbol{c}_{i:j}$ denotes the i-th to j-th sentences. $\boldsymbol{c}_{<i}$ and $\boldsymbol{c}_{>i}$ respectively denote the preceding and subsequent context of $\boldsymbol{c}_i$. Given contexts $\boldsymbol{c}_{<i}$ and $\boldsymbol{c}_{>i}$, we first train a context-LM by maximizing $p(\boldsymbol{c}_i|\boldsymbol{c}_{<i}, \boldsymbol{c}_{>i})$. The input is a sequence of words and the input representation for each word is the addition of three embeddings: the sentence-position embedding, token-position embedding and the word embedding. Predicting $\boldsymbol{c}_i$ follows a word-by-word fashion. We consider the style of both left-to-right generation and right-to-left generation to optimize $p(\boldsymbol{c}_i|\boldsymbol{c}_{<i}, \boldsymbol{c}_{>i})$, which is respectively given by the following objective:

$$p(\overrightarrow{\boldsymbol{c}}_i|\boldsymbol{c}_{<i}, \boldsymbol{c}_{>i}) = \prod_{j=1}^{n} p(w_{i,j}|\boldsymbol{c}_{<i}, \boldsymbol{c}_{>i}, \boldsymbol{w}_{i,<j})$$
$$p(\overleftarrow{\boldsymbol{c}}_i|\boldsymbol{c}_{<i}, \boldsymbol{c}_{>i}) = \prod_{j=n}^{1} p(w_{i,j}|\boldsymbol{c}_{<i}, \boldsymbol{c}_{>i}, \boldsymbol{w}_{i,>j}) \quad (1)$$

$p(\boldsymbol{c}_i|\boldsymbol{c}_{<i}, \boldsymbol{c}_{>i})$ models the forward probability from contexts to sentences. For two sentences of the same meaning, the probability of generating contexts given the two sentences should be also the same, which correspond to the backward probability given from sentences to contexts. This is akin to the bi-directional mutual-information based generation strategy (Fang et al., 2015; Li et al., 2016a; Li and Jurafsky, 2016; Wang et al., 2021). The backward probability can be modeled by predicting preceding contexts given subsequent contexts $p(\boldsymbol{c}_{<i}|\boldsymbol{c}_i, \boldsymbol{c}_{>i})$ and to predict subsequent contexts given preceding contexts $p(\boldsymbol{c}_{>i}|\boldsymbol{c}_{<i}, \boldsymbol{c}_i)$.

We implement the above models, i.e. $p(\overrightarrow{\boldsymbol{c}}_i|\boldsymbol{c}_{<i}, \boldsymbol{c}_{>i})$, $p(\overleftarrow{\boldsymbol{c}}_i|\boldsymbol{c}_{<i}, \boldsymbol{c}_{>i})$, $p(\boldsymbol{c}_{<i}|\boldsymbol{c}_i, \boldsymbol{c}_{>i})$, $p(\boldsymbol{c}_{>i}|\boldsymbol{c}_{<i}, \boldsymbol{c}_i)$ based on the SEQ2SEQ structure on a subset of CommonCrawl containing 10 billion tokens in total. We use Transformers as the backbone (Vaswani et al., 2017)[2] with the number of encoder blocks, decoder blocks, the number of heads, d_{model} and $d_f f$ set to 6, 6, 8, 512 and 2048. We use adam (Kingma and Ba, 2014) for optimization, with learning rate of 1e-4, β_1 = 0.9, β_2 = 0.999. We consider a maximum number of +800 and -800 tokens as contexts.

3.2 Paraphrase Candidate Generation

Using the pretrained *context-LM* models, we generate potential paraphrases by decoding multiple outputs given the input sentence only based on $p(\overrightarrow{\boldsymbol{c}}_i|\boldsymbol{c}_{<i}, \boldsymbol{c}_{>i})$. The other three contextual objectives, i.e., $p(\overleftarrow{\boldsymbol{c}}_i|\boldsymbol{c}_{<i}, \boldsymbol{c}_{>i})$, $p(\boldsymbol{c}_{<i}|\boldsymbol{c}_i, \boldsymbol{c}_{>i})$ and $p(\boldsymbol{c}_{>i}|\boldsymbol{c}_{<i}, \boldsymbol{c}_i)$ cannot be readily used at the decoding stage since their computations require the completion of the target generation. They will thus be used at the later reranking stage. We use diverse decoding strategy of beam search (Li et al., 2016b) to generate diverse candidates. Decoded candidates are guaranteed to be fluent.[3]

3.3 Paraphrase Filtering

The decoded andidates can not be readily used since (1) candidates often differ only by punctuation or minor morphological variations, with almost all words overlapping, and (2) many of them are not of the same meaning. We thus propose to further rank a candidate pairs. The ranking model consists of three parts:

[2]The four models share the same structure but with a special objective-specific token appended to the model input notifying different objectives.

[3]Implementation-wise, we first cache all the possible candidate paraphrase pairs for all input context sentences. These pairs are then used for filtering, as will be detailed in the next section. We also impose a constraint that at most one paraphrase pair with respect to an input context is selected for training the final SEQ2SEQ model (Section 3.4).

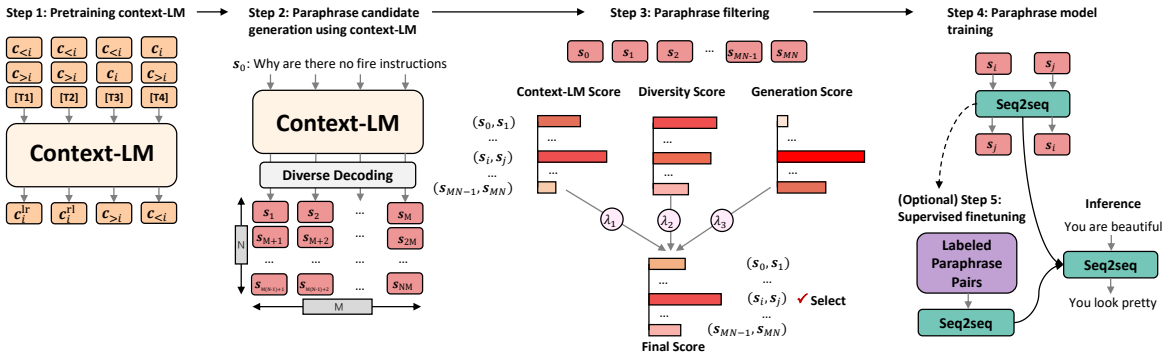

Figure 1: An overview of the proposed ConRPG framework. Step 1: we first train a *context-LM* model that predicts the sentence probability in an autoregressive manner given contexts. Step 2: the *context-LM* model is used to decode multiple candidate paraphrases with respect to a given context using diverse decoding of beam search. Step 3: paraphrase candidates are filtered based on different scoring functions, i.e., the context-LM score, the diversity score and the generation score. Step 4: the selected pair is used to train a SEQ2SEQ model, which can be latter used for supervised finetuning or be directly used for unsupervised paraphrase generation.

3.3.1 Context LM Score

For a pair of sentences $\boldsymbol{s}_1$ and $\boldsymbol{s}_2$ of the same meaning, differences between the probabilities of generating them given the same context should be very similar. In the same way, the probabilities of predicting left and right contexts given the two sentences with the same meaning should also be similar. The ranking scoring function to rank $(\boldsymbol{s}_1, \boldsymbol{s}_2)$ consists the following parts: (1) the probability difference in generating two sentences given contexts, i.e., $\frac{1}{|\boldsymbol{s}|}\log p(\overrightarrow{\boldsymbol{s}}|\boldsymbol{c}_{<i}, \boldsymbol{c}_{>i})$ and $\frac{1}{|\boldsymbol{s}|}\log p(\overleftarrow{\boldsymbol{s}}|\boldsymbol{c}_{<i}, \boldsymbol{c}_{>i})$; (2) the probability difference in generating contexts given two sentences, i.e., $\frac{1}{|\boldsymbol{c}_{<i}|}|\log p(\boldsymbol{c}_{<i}|\boldsymbol{s}, \boldsymbol{c}_{>i})$ and $\frac{1}{|\boldsymbol{c}_{<i}|}|\log p(\boldsymbol{c}_{<i}|\boldsymbol{s}, \boldsymbol{c}_{>i})$.

3.3.2 Lexicon and Syntactic Diversity

Two identical sentences will have the optimal score, which does not serve our purpose since we wish paraphrases to be as diverse as possible (Li et al., 2018). We consider two types of diversity: (1) lexicon diversity, which encourages individual word or phrase replacements using synonyms; and (2) syntactic diversity, which encourages syntactic shifting such as heavy NP shift. Lexicon diversity is measured by the unigram-based Jaccard distance between two sentences. Syntactic diversity is measured by the relative position change for shared unigrams. If $\boldsymbol{s}_2$ contains multiple copies of a word w in $\boldsymbol{s}_1$, we pick the nearest copy. Let $\text{pos}_{\boldsymbol{s}}(w)$ denote the position index of w in $\boldsymbol{s}$. The combination of lexicon and syntactic diversity is given as follows:

$$\text{S}_{\text{diversity}}(\boldsymbol{s}_1, \boldsymbol{s}_2) = \beta_1 \frac{|\boldsymbol{s}_1 \cap \boldsymbol{s}_2|}{|\boldsymbol{s}_1 \cup \boldsymbol{s}_2|} + \beta_2 \frac{1}{|\boldsymbol{s}_1 \cap \boldsymbol{s}_2|} \sum_{w \in \boldsymbol{s}_1 \cap \boldsymbol{s}_2} \frac{|\text{pos}_{\boldsymbol{s}_1}(w) - \text{pos}_{\boldsymbol{s}_2}(w)|}{\max(|\boldsymbol{s}_1|, |\boldsymbol{s}_2|)} \quad (2)$$

where the first part denotes the unigram Jaccard distance, and the second part denotes the relative position change for unigrams.

3.3.3 Mutual Generation Score

It is noteworthy that an intrinsic drawback of the proposed methodology (and other paraphrase generation methods as well) is that, two sentences that can fit into the same context are not necessarily of the exactly same meaning, e,g, sentences with very similar general semantics but vary in some specific details (e.g., number). Think about two sentences, *I spent 5 dollars on this mug.* v.s. *I spent 6 dollars on this mug*. If one sentence fits into certain contexts, it is very likely that the other sentence will also fit in. The issue can be alleviated with more contexts considered, but the practical problem still remains because our model can only consider a very limited number of contexts due to hardware limitations.

We propose a strategy to address this drawback. The strategy is inspired by the famous idiom that "*Happy families are all alike; every unhappy family is unhappy in its own way*". Paraphrases share the same meaning in the vector space, and there should be a direct and easy mapping between them. Non-paraphrases are different in random ways. It is thus easier to predict a paraphrase given a sentence than

predict a specific non-paraphrase given the sentence. For example, p("six dollars"|"6 dollars") should be higher than generating a random sentence give the sentence e.g., p("5 dollars"|"6 dollars"). This is because, there are so many ways to generate non-paraphrase e.g., p("5 dollars"|"6 dollars") and p("7 dollars"|"6 dollars"), etc. These non-paraphrases split the probability, making the probability for an individual non-paraphrase low. To this end, we train a SEQ2SEQ model (Sutskever et al., 2014) on 8 million pairs of decoded candidates using Transformer-based. Next, using this model, we give the mutual decoding score for any sentence pair $(\boldsymbol{s}_1, \boldsymbol{s}_2)$ as follows:

$$\mathrm{S}_{\text{generation}} = \gamma_1 \frac{1}{|\boldsymbol{s}_1|} \log p(\boldsymbol{s}_1|\boldsymbol{s}_2) + \gamma_2 \frac{1}{|\boldsymbol{s}_2|} \log p(\boldsymbol{s}_2|\boldsymbol{s}_1) \quad (3)$$

For a sentence pair of the same meaning, they should have higher values of Eq.3.

3.3.4 Final Ranking Model

The final ranking score is a linear combination of scores above as follows:

$$\begin{aligned} &\mathrm{S}(\boldsymbol{s}_1, \boldsymbol{s}_2) = \mathrm{S}_{\text{context}}(\boldsymbol{s}_1, \boldsymbol{s}_2) \\ &+ \mathrm{S}_{\text{diversity}}(\boldsymbol{s}_1, \boldsymbol{s}_2) + \mathrm{S}_{\text{generation}}(\boldsymbol{s}_1, \boldsymbol{s}_2) \end{aligned} \quad (4)$$

We build a ranking model to learn weights (i.e., α, β, γ, eight parameters in total). To train the ranking model, we annotate a small proportion of data on Amazon Mechanical Turk. A Turker is first given a sentence (denoted by a) randomly picked from the candidate pool. Next, the Turker is given two other decoded sentences (b_1 and b_2), and is asked to decide which one is a better paraphrase of a, in terms of three aspects: (1) semantics: whether the two sentences are of 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 and will be further removed. We labeled a total number of 2K pairs. Let b_+ denote the better paraphrase by annotators, and b_- denote the other. Based on the labeled dataset, a simple pairwise ranking model (Liu, 2011) is built for weight learning:

$$L = \max(0, 1 + S(a, b_+) - S(a, b_-)) \quad (5)$$

It is worth noting that the filtering module provides a channel for developers to intervene with the data generation process, as developers can develop their own scoring functions to generate paraphrases of specific features. This leads to a more controllable paraphrase model.

3.4 Paraphrase Model Training

We select 10 million paraphrase pairs in total based on criteria above, on which we train a SEQ2SEQ model for paraphrase generation, using one sentence of the pair as the input, and the other as the output. We use the Transformer-base (Vaswani et al., 2017) as the model backbone. We use Adam (Kingma and Ba, 2014) with learning rate of 1e-4, $\beta_1 = 0.9$, $\beta_2 = 0.98$ and a warmup step of 4K. The trained model can be directly used for paraphrase generation in the unsupervised setup (Roy and Grangier, 2019; Liu et al., 2019).

For the supervised setup (Witteveen and Andrews, 2019; Kazemnejad et al., 2020; Hegde and Patil, 2020), 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 model on the supervised paraphrase pairs, where we initialize the model using the pre-trained model, and run additional iterations on the supervised dataset. 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 hyperparameters, to be tuned on the dev set.

It is worth nothing that the SEQ2SEQ model here is different from the SEQ2SEQ model in the filtering stage, as the model here is trained on the remaining paraphrase pairs and used for direct paraphrase generation, while the other is trained on the noisy pairs and used for candidate filtering.

4 Experiments

4.1 Datasets

We carry out experiments in both supervised and unsupervised setups. For the unsupervised setting, we use the Quora, Wikianswers (Fader et al., 2013), MSCOCO (Lin et al., 2014) and Twitter (Lan et al., 2017) datasets. For the supervised setting, we use the Quora and Wikianswers datasets.

- **Quora**: The Quora question pair dataset[4] contains 140K parallel paraphrases and 260K non-parallel sentences. We follow the standard setup in Miao et al. (2019) where 3K and 30K paraphrase pairs are respectively used for validation and test.

[4] https://www.kaggle.com/c/quora-question-pairs

- **Wikianswers**: The Wikianswers dataset (Fader et al., 2013) contains 2.3M paraphrase pairs extracted from the Wikianswers website. We follow Liu et al. (2019) to randomly pick 5K pairs for validation and 20K for test.[5]
- **MSCOCO**: The MSCOCO dataset (Lin et al., 2014) contains over 500K paraphrase pairs for 120K image captions. We follow the standard dataset split and the evaluation protocol in Liu et al. (2019).
- **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. (2019).

4.2 Baselines and Metrics

We compare our proposed ConRPG model to the following existing paraphrase generation models. Unsupervised paraphrase generation baselines we consider include:

- **VAE**: paraphrases are sampled by encoding a sentence to a continuous space using (VAEs) (Bowman et al., 2016).
- **Lag VAE**: A sophisticated version of VAE to deal with the posterior collapse issue He et al. (2019).
- **CGMH**: Miao et al. (2019) used Metropolis–Hastings sampling for constrained sentence generation, where a word can be deleted, replaced or inserted into the current sentence based on the sampling distribution.
- **UPSA**: Liu et al. (2019) proposed to treat unsupervised paraphrase generation as an optimization problem with an objective combining semantic similarity, expression diversity and language fluency being optimized using simulated annealing.
- **Corruption**: (Hegde and Patil, 2020) proposed strategy of corrupting input sentences by removing stop words and randomly shuffle and replace the remaining 20% words. We use BART (Lewis et al., 2019) as the backbone to generate targets given corrupted inputs.

Results for VAE, Lag VAE, CGMH and UPSA on different datasets are copied from Miao et al. (2019) and Liu et al. (2019). Supervised paraphrase generation baselines include:

- **ResidualLSTM**: Prakash et al. (2016b) deepened the LSTM network by stacking multiple layers with residual connection.
- **VAE-SVG-eq**: Gupta et al. (2018) combined VAEs with LSTMs for paraphrase generation. Both encoder and decoder are conditioned on the source input sentence so that more consistent paraphrases can be generated.
- **Pointer**: See et al. (2017) augmented the standard SEQ2SEQ model by using a pointer mechanism which can copy source words in the input rather than decode from scratch.
- **Transformer**: Vaswani et al. (2017) proposed the Transformer architecture which is based on the self-attention mechanism.
- **DNPG**: Li et al. (2019) proposed a Transformer-based model that can learn and generate paraphrases at different granularities.

Results for ResidualLSTM, VAE-SVG-eq, Pointer, Transformer on various datasets are copied from Li et al. (2019). For reference purposes, we also implement the **BT** baseline inspired by the idea of back-translation (Sennrich et al., 2016; Wieting et al., 2017). We use Transformer-large as the backbone. BT is trained end-to-end on WMT'14 En↔Fr.[6] 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.

We evaluate all models using BLEU (Papineni et al., 2002), iBLEU (Sun and Zhou, 2012) and ROUGE scores (Lin, 2004) . The iBLEU score penalizes the similarity of the generated paraphrase with respect to the original input sentence. Concretely, the iBLEU score of a triple of sentences $(\boldsymbol{s}, \boldsymbol{r}, \boldsymbol{c})$ is given by:

$$\begin{aligned} \text{iBLEU}(\boldsymbol{s}, \boldsymbol{r}, \boldsymbol{c}) = \alpha \text{BLEU}(\boldsymbol{c}, \boldsymbol{r}) \\ - (1-\alpha)\text{BLEU}(\boldsymbol{c}, \boldsymbol{s}) \end{aligned} \tag{6}$$

where $\boldsymbol{s}$ is the input sentence, $\boldsymbol{r}$ is the reference paraphrase and $\boldsymbol{c}$ is generated paraphrase. α is set to 0.8 following prior works.

4.3 In-domain Results

We first show the in-domain results in Table 1. As can be seen, across all datasets, the proposed ConRPG model significantly outperforms baselines in

[5]Note that the selected data is different from Liu et al. (2019) but is comparable in the statistical sense.

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

both supervised and unsupervised settings. For the supervised setting, ConRPG yields an approximately 2-point gain across different evaluation metrics against the strong DNPG baseline on both Quora and Wikianswers. We also observe that the BT model is able to achieve competitive results. This shows that back-translation can serve as a simple yet strong baseline for paragraph generation. For the unsupervised setting, we observe substantial performance boosts brought by ConRPG over existing unsupervised methods including the state-of-the-art model UPSA. It is also surprising to see that unsupervised ConRPG outperforms the supervised VAE-SVG-eq model and achieves comparable results to supervised baselines such as Transformer.

4.4 Domain-adapted Results

We test the domain adaptation ability of the proposed method on the Quora and Wikianswers datasets. Results are shown in Table 3. We can see that ConRPG significantly outperforms baselines in both settings, i.e. *Quora→Wikianswers* and *Wikianswers→Quora*, showing the better ability of ConRPG for domain adaptation.

4.5 Human Evaluation

To further validate the performance of the proposed model, we sample 400 sentences from the Quora test set for human evaluation. We assign the input sentence and its generated paraphrase to three human annotators at Amazon Mechanical Turk (AMT), with "> 95% HIT approval rate". Turkers are asked to evaluate the quality of generated paraphrases by considering three aspects semantics, diversity and fluency, as detailed in Section 3.3.4. Each paraphrase is labeled by a 5-point scale (Strongly Agree, Agree, Unsure, Disagree, Strongly Disagree) and assigned to three annotators. We evaluate three models: BT, Corruption, and the proposed ConRPG model. The Cohen's kappa score (McHugh, 2012) for the three aspects are 0.55, 0.52 and 0.49, indicating moderate inter-annotator agreement. Table 2 presents the human evaluation results. As can be seen from the table, the proposed ConRPG model significantly outperforms BT and Corruption in terms of all three aspects, which is consistent with the automatic evaluation results.

	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
	Transformer+Copy	17.98	24.77	63.34	37.31
	DNPG	18.01	25.03	63.73	37.75
	BT	17.73	24.99	62.07	36.12
	ConRPG	**19.96**	**26.81**	**65.03**	**38.49**
		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
	Transformer+Copy	31.43	37.88	55.88	23.37
	DNPG	34.15	41.64	57.32	25.88
	BT	33.65	39.70	56.89	25.22
	ConRPG	**35.28**	**42.25**	**58.40**	**26.44**
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	11.64	11.59	58.20	32.04
	Corruption	12.32	17.97	59.14	32.44
	ConRPG	**12.68**	**18.31**	**59.62**	**33.10**
		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	24.17	31.75	53.69	20.63
	Corruption	24.40	32.05	53.77	21.22
	ConRPG	**25.98**	**32.89**	**54.65**	**22.25**
		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	9.72	14.36	37.64	11.81
	Corruption	10.32	15.60	38.12	12.40
	ConRPG	**11.17**	**16.98**	**39.42**	**13.50**
		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	5.11	6.99	29.11	8.95
	Corruption	5.32	7.11	29.80	9.32
	ConRPG	**5.83**	**7.32**	**30.81**	**10.08**

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

5 Ablation Study

5.1 Size of Data to Train *context-LM*

First, we would like to understand how the data size for training *context-LM* effects the downstream performance of Wikianswers. Table 5 presents the results where the training data size is respectively 10M, 100M, 1B and 10B tokens. We can observe that with more training data, downstream performances under both setups increase. This is because more training data leads to a more reliable context regularization, and thus the trained model can produce paraphrases with higher qualities.

Model	Semantics	Diversity	Fluency
ConRPG	**3.78 (0.5)**	**4.01 (0.4)**	**4.21 (0.3)**
Corruption	3.14 (0.6)	3.17 (0.5)	4.19 (0.4)
BT	3.04 (0.6)	3.32 (0.5)	3.89 (0.4)

Table 2: Human evaluation results for BT, UPSA and ConRPG under the unsupervised setup.

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	12.54	17.98	59.43	32.54
ConRPG	**13.25**	**19.28**	**60.55**	**34.17**
	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	26.11	35.28	57.29	23.88
ConRPG	**28.14**	**37.93**	**57.98**	**25.32**

Table 3: Domain-adapted performances.

5.2 Context Length to Train *context-LM*

Table 6 presents the influence of context length used to train *context-LM* on Wikianswers. As can be seen, the performance is sensitive to the context length, which can be explained by the fact that more contexts lead to a significantly better language modeling.

5.3 Percentage of Selected Paraphrase Pairs

Table 7 presents the impact of the percentage of selected paraphrase pairs in the filtering process on the final performance of Wikianswers. We tune the ratio ρ, which is defined as the number of remaining paraphrase pairs divided by the number of input contexts for *context-LM*. $\rho = 1$ is what we use in this work: selecting the top-1 paraphrase pair for each input context makes the number of remaining pairs equal to the number of input contexts. As expected, either too few or too many selected paraphrase pairs leads to worse performances. Too few pairs lead to insufficient training and too many pairs lead to noise that harm the final performance. A tricky balance of the percentage of selected paraphrase pairs is thus crucial for better final performances.

5.4 Effects of Different Modules

We are interested in the effectiveness of each module within the proposed framework. Table 8 shows the performance:

(1) Removing the entire *filtering* module leads to the most degradation in performance, which is in line with our expectation: with filtering, high quality paraphrase pairs that both share the same meaning and are diverse in lexicon can be selected for training the final paraphrase generation model.

(2) Removing *backward*, i.e., $p(\boldsymbol{c}_{<i}|\boldsymbol{c}_i, \boldsymbol{c}_{>i})$ and $p(\boldsymbol{c}_{>i}|\boldsymbol{c}_{<i}, \boldsymbol{c}_i)$, leads to the second largest performance reduction. This is because removing *backward* greatly weakens the strength of context regularization, introducing more noise for the subsequent paraphrase filtering phase.

(3) Removing *right-to-left*, i.e., $p(\overleftarrow{\boldsymbol{c}}_i|\boldsymbol{c}_{<i}, \boldsymbol{c}_{>i})$, leads to a slight drop in performance.

(4) Removing the diversity score or the generation score harms model performances. This observation verifies that using scores from different aspects significantly helps paraphrase quality.

6 Conclusion

In this paper, we propose ConRPG, a paradigm for paraphrase generation using context regularizer. ConRPG is based on the assumption that the probabilities of generating two sentences with the same meaning based on the same context should be the same. We acknowledge that the current system is rather complicated, which requires multiple pipelines and modules to build. We will simplify the system in future work.

Acknowledgement

This work was supported by Key-Area Research and Development Program of Guangdong Province (No. 2019B121204008). We thank the High-Performance Computing Platform at Peking University and the PCNL Cloud Brain for providing platforms of data analysis and model training.

References

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

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

Yu Bao, Hao Zhou, Shujian Huang, Lei Li, Lili Mou, Olga Vechtomova, Xin-yu Dai, and Jiajun Chen. 2019. Generating sentences from disentangled syntactic and

Input	Corruption	ConRPG
What should be the first computer table language I learn?	What should be my first computer table language?	If I want to learn a programming language, which one should I learn first?
How do I overcome my shyness with women?	How do I overcome shyness with women?	How can I overcome being shy when women are around.
Should Harry Potter have ended up with Cho Chang?	Should Harry Potter be with Chang Cho?	Should Harry Potter and Cho Chang end up being together?
How do I become a data scientist in Malaysia?	How can I become a scientist in Malaysia?	What do I need to do if I want to become a data scientist in Malaysia?
What are the rate common regrets in old age?	What are the common regrets in old age?	What do old people regret the most?

Table 4: Sampled paraphrases from the Corruption model and the ConRPG model.

Setup	10M	100M	1B	10B
Unsupervised	13.8	19.2	24.9	26.0
Supervised	31.4	32.5	34.4	35.3

Table 5: The effect of data size for training *context-LM*. The iBLEU score is reported on Wikianswers.

Setup	100	400	800
Unsupervised	21.9	25.1	26.0
Supervised	31.6	34.1	35.3

Table 6: The effect of context length for *context-LM*. The iBLEU score is reported on Wikianswers.

Setup/iBLEU	ρ=0.01	ρ=0.1	ρ=1	ρ=5
Unsupervised	20.8	25.0	26.0	24.2
Supervised	32.5	34.8	35.3	34.8

Table 7: The effect of percentage of selected candidates for candidate reranking on Wikianswers.

Model	iBLEU
Full	25.98
w/o filtering	20.12 (-5.86)
w/o backward	23.29 (-2.69)
w/o right-to-left	25.56 (-0.42)
w/o diversity	23.99 (-1.99)
w/o generation	24.80 (-1.18)

Table 8: The effect of different modules within ConRPG. *w/o filtering* means removing the filtering phase and randomly choosing a paraphrase pair for each context. *w/o backward* means removing the $p(\boldsymbol{c}_{<i}|\boldsymbol{c}_i, \boldsymbol{c}_{>i})$ and $p(\boldsymbol{c}_{>i}|\boldsymbol{c}_{<i}, \boldsymbol{c}_i)$ training objectives. *w/o right-to-left* means removing the $p(\overleftarrow{\boldsymbol{c}}_i|\boldsymbol{c}_{<i}, \boldsymbol{c}_{>i})$ training objective. *w/o diversity* means removing the diversity score and *w/o generation* means removing the generation score.

semantic spaces. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 6008–6019, Florence, Italy. Association for Computational Linguistics.

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.

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.

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

David Chen and William B Dolan. 2011. Collecting highly parallel data for paraphrase evaluation. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 190–200.

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

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.

Hao Fang, Saurabh Gupta, Forrest Iandola, Rupesh Srivastava, Li Deng, Piotr Dollár, Jianfeng Gao, Xiaodong He, Margaret Mitchell, John C. Platt, C. Lawrence Zitnick, and Geoffrey Zweig. 2015. From captions to visual concepts and back.

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.

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.

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*.

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.

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

Amirhossein Kazemnejad, Mohammadreza Salehi, and Mahdieh Soleymani Baghshah. 2020. Paraphrase generation by learning how to edit from samples. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6010–6021, Online. Association for Computational Linguistics.

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*.

Kalpesh Krishna, John Wieting, and Mohit Iyyer. 2020. Reformulating unsupervised style transfer as paraphrase generation. *arXiv preprint arXiv:2010.05700*.

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.

Mike Lewis, Marjan Ghazvininejad, Gargi Ghosh, Armen Aghajanyan, Sida Wang, and Luke Zettlemoyer. 2020. Pre-training via paraphrasing. *arXiv preprint arXiv:2006.15020*.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Ves Stoyanov, and Luke Zettlemoyer. 2019. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. *arXiv preprint arXiv:1910.13461*.

Jiwei Li, Michel Galley, Chris Brockett, Jianfeng Gao, and Bill Dolan. 2016a. A diversity-promoting objective function for neural conversation models. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 110–119, San Diego, California. Association for Computational Linguistics.

Jiwei Li and Dan Jurafsky. 2016. Mutual information and diverse decoding improve neural machine translation. *arXiv preprint arXiv:1601.00372*.

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

Zichao Li, Xin Jiang, Lifeng Shang, and Hang Li. 2018. 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.

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.

Tie-Yan Liu. 2011. *Learning to rank for information retrieval*. Springer Science & Business Media.

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

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.

Mary L McHugh. 2012. Interrater reliability: the kappa statistic. *Biochemia medica: Biochemia medica*, 22(3):276–282.

Kathleen R. McKeown. 1979. Paraphrasing using given and new information in a question-answer system. In *17th Annual Meeting of the Association for Computational Linguistics*, pages 67–72, La Jolla, California, USA. Association for Computational Linguistics.

Marie Meteer and Varda Shaked. 1988. Strategies for effective paraphrasing. In *Coling Budapest 1988 Volume 2: International Conference on Computational Linguistics*.

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.

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.

Chris Quirk, Chris Brockett, and William Dolan. 2004. Monolingual machine translation for paraphrase generation. In *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing*, pages 142–149, Barcelona, Spain.

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.

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*.

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. 2021. Sentence similarity based on contexts. *arXiv preprint arXiv:2105.07623*.

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.

Shuhe Wang, Yuxian Meng, Xiaofei Sun, Fei Wu, Rongbin Ouyang, Rui Yan, Tianwei Zhang, and Jiwei Li. 2021. Modeling text-visual mutual dependency for multi-modal dialog generation. *arXiv preprint arXiv:2105.14445*.

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 embed-

dings 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*.

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.