Dictionary-Guided Editing Networks for Paraphrase Generation

Shaohan Huang,[†§] **Yu Wu,**[‡] **Furu Wei,**[†] **Zhongzhi Luan**[§]
[§]Sino-German Joint Software Institute, Beihang University, Beijing, China
[†]Microsoft Research, Beijing, China
[‡]State Key Lab of Software Development Environment, Beihang University, Beijing, China
{shaohanh, fuwei}@microsoft.com wuyu@buaa.edu.cn

Abstract

An intuitive way for a human to write paraphrase sentences is to replace words or phrases in the original sentence with their corresponding synonyms and make necessary changes to ensure the new sentences are fluent and grammatically correct. We propose a novel approach to modeling the process with dictionary-guided editing networks which effectively conduct rewriting on the source sentence to generate paraphrase sentences. It jointly learns the selection of the appropriate word level and phrase level paraphrase pairs in the context of the original sentence from an off-the-shelf dictionary as well as the generation of fluent natural language sentences. Specifically, the system retrieves a set of word level and phrase level paraphrase pairs derived from the Paraphrase Database (PPDB) for the original sentence, which is used to guide the decision of which the words might be deleted or inserted with the soft attention mechanism under the sequence-to-sequence framework. We conduct experiments on two benchmark datasets for paraphrase generation, namely the MSCOCO and Quora dataset. The automatic evaluation results demonstrate that our dictionary-guided editing networks outperforms the baseline methods. On human evaluation, results indicate that the generated paraphrases are grammatically correct and relevant to the input sentence.

Introduction

Paraphrase generation aims to generate restatements of the meaning of a text or passage using other words. It is a fundamental task in natural language processing with many applications in information retrieval, question answering, dialogue, and conversation systems. Existing work on paraphrase generation focuses on generating paraphrase sentences from scratch. Traditional paraphrase generation methods have been addressed using rule-based approaches (Hassan et al. 2007; Zhao et al. 2009) and statistical machine translation (SMT) based approaches (Quirk, Brockett, and Dolan 2004; Zhao et al. 2009; 2010). Recently, neural networks based generative models under the sequence-to-sequence framework have also been used for paraphrase generation (Prakash et al. 2016; Gupta et al. 2018).

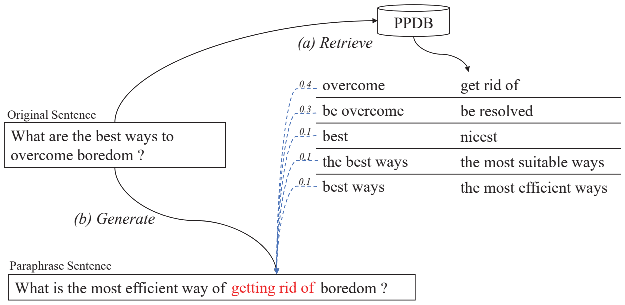

Figure 1: The dictionary-guided editing networks model first retrieves a group of paraphrase pairs from the Paraphrase Database and then generates a paraphrase using the original sentence as a prototype.

However, an intuitive way for a human to write paraphrase sentences is to replace words or phrases in the original sentence with their corresponding synonyms and make necessary changes to ensure the new sentences are fluent and grammatically correct. Figure 1 shows an example. Given the input sentence "*What are the best ways to overcome boredom?*", we can first replace "*overcome*" with the word level paraphrase phrases "*get rid of*", and then make small changes over the new sentence to ensure it is grammatically correct and fluent. Certainly, it should be emphasized that the selection of context-relevant paraphrase pairs from an off-the-shelf dictionary with respect to the original sentence is also important for a good revision. This process demonstrates that humans usually write paraphrase sentences by editing the input sentence, which motivates us to develop models for paraphrase generation through editing.

We are inspired by Guu et al.'s pioneer work (2018) on a new paradigm to generate sentences. Specifically, they propose a new generative model of sentences that first samples a prototype sentence from the training corpus and then edits it into a new sentence using a randomly sampled edit vector. For prototypes in paraphrase generation task, we use original sentences as prototypes directly instead of random examples. For edit vectors, unlike randomly sampling the edit vector to generate a new sentence, we can leverage the word level and phrase level paraphrase pairs (e.g. synonyms) to construct the editing vector where the deletion of words from the original sentence and the insertion words into the target sentence can be explicitly modeled.

In this paper, we propose a dictionary-guided editing networks for paraphrase generation which effectively conducts rewriting on the source sentence to generate paraphrase sentences. It jointly learns the selection of the appropriate word level and phrase level paraphrase pairs in the context of the original sentence from an off-the-shelf dictionary as well as the generation of fluent natural language sentences. As shown in Figure 1, for the original sentence "*What are the best ways to overcome boredom?*", the system first retrieves a set of word level and phrase level paraphrase pairs, which is derived from the Paraphrase Database (PPDB) (Pavlick et al. 2015). We expect that the paraphrase pairs can guide the decision on which the words might be deleted or inserted. We leverage the paraphrase pairs to construct the editing vector with the soft attention mechanism. The editing vector is computed by the weighted sum of insertion word embeddings and deletion words in the dictionary. For instance, we wish pairs ("*overcome*", "*get rid of*") and ("*be overcome*", "*be resolved*") get larger weights than others when we revise the word "*overcome*". The deletion of words from the original sentence and the insertion words into the target sentence can be explicitly modeled through the soft attention mechanism.

We conduct experiments on the benchmark MSCOCO and Quora datasets for paraphrase generation. We compare our dictionary-guided editing networks with sequence-to-sequence generation baselines including the state-of-the-art variational autoencoder model (Gupta et al. 2018) for paraphrase generation. The automatic evaluation results demonstrate that the dictionary-guided editing networks outperforms existing sequence-to-sequence generation baselines and achieves state-of-the-art results, whereas human evaluation results indicate that our generated paraphrases are grammatically correct, fluent and relevant to the input sentence.

The rest of this paper is organized as follows: In Section 2 we show the detailed design of our dictionary-guided editing network model. In Section 3 we conduct paraphrase generation experiments on two datasets and demonstrate the evaluation results. Section 4 gives a brief overview of the recent history of paraphrase generation and presents a description of text editing methods. Section 5 concludes this paper and outlines future work.

Methodology

Problem Statement and Model Overview

Suppose that we have a parallel data set $\mathcal{P} = \{(X_i, Y_i)\}_{i=1}^{L}$, where (X_i, Y_i) is a pair of sentence-level paraphrases. Furthermore, we have access to a dictionary $\mathcal{C} = \{(o_i, p_i)\}_{i=1}^{N}$ that consists of a huge amount of word-level and phrase-level paraphrases, where o_i and p_i are either a word or a phrase. Given an original sentence X_i, the dictionary can help us to find possible word substitutions in the paraphrasing process. Our goal is to learn a paraphrase generation model with the use of $\mathcal{C}$ and $\mathcal{P}$. In the following parts, we will first introduce how to find possible word level or phrase level paraphrase pairs $\mathcal{D}$ from $\mathcal{C}$, and then we present how to generate a fluent paraphrasing Y with X and $\mathcal{D}$.

The overview of our model is shown in Figure 2. Given a sentence $X_i \in \mathcal{P}$, we first retrieve a set of word level and phrase level paraphrase pairs $\mathcal{D} = \{(o_i, p_i)\}_{i=1}^{M}$ derived from paraphrase corpus $\mathcal{C}$. Secondly, we learn the dictionary-guided editing networks model to generate the paraphrase sentence Y with the original sentence X and the paraphrase pairs $\mathcal{D}$ as input.

Retrieval

Our model relies on the observation that humans usually write paraphrase sentences by replacing words or phrases in the original sentence with their corresponding synonyms. Therefore, the first step of our method is to retrieve a set of lexical or phrasal paraphrase pairs for the original sentence. For example, for original sentence X "*What are the best ways to overcome boredom*", we can find some paraphrase pairs such as ("*overcome*", "*get rid of*"), ("*the best ways*", "*the most suitable ways*"), and ("*the best ways*", "*the most efficient ways*").

Our system retrieves word level and phrase level paraphrase pairs derived from the Paraphrase Database (PPDB) (Pavlick et al. 2015). PPDB is an automatically extracted database containing millions of paraphrases in different languages. It contains three types of paraphrases: lexical (single word to single word), phrasal (multiword to single/multiword), and syntactic (paraphrase rules containing non-terminal symbols). We use PPDB with the lexical and phrasal types as raw paraphrased corpus $\mathcal{C}$.

We construct the paraphrase pairs $\mathcal{D} = \{(o_i, p_i)\}_{i=1}^{M}$ from the off-the-shelf dictionary $\mathcal{C}$ where the size of dictionary $\mathcal{D}$ is M. We leverage Lucene[1] to index the paraphrase pair corpus $\mathcal{C}$. Specifically, we index all original words and phrases in corpus $\mathcal{C}$. The retrieval strategy consists of two steps. We first retrieve top $10 \times M$ paraphrase pairs as candidates for the original sentence X using the default ranking function in Lucene. Then, we rank these candidates by combining the TF-IDF weighted word overlap and the PPDB score. The ranking score of pair (o_i, p_i) is formulated as:

$$score = \sum_{w \in o_i \cap X} tf_w \cdot idf_w + score_p(o_i, p_i) \tag{1}$$

where $score_p(o_i, p_i)$ is the PPDB score of pair (o_i, p_i), which is computed by a regression model in PPDB (Pavlick et al. 2015). For the original sentence X, we take the top M word level or phrase level paraphrase pairs as $\mathcal{D}$. In the PPDB, one word/phrase may correspond to several paraphrased words or phrases. In other words, different paraphrasing pairs in PPDB may share the same left-hand side word/phrase. In order to improve the diversity of paraphrase pairs $\mathcal{D}$, we will only keep the one with the highest ranking score calculated by PPDB.

Dictionary-Guided Editing

After finding the off-the-shelf paraphrase pairs $\mathcal{D} = \{(o_i, p_i)\}_{i=1}^{M}$ for original sentence X, we implement a dictionary-guided editing networks model to generate the paraphrase sentence Y by revising the original sentence X.

[1] https://lucene.apache.org/

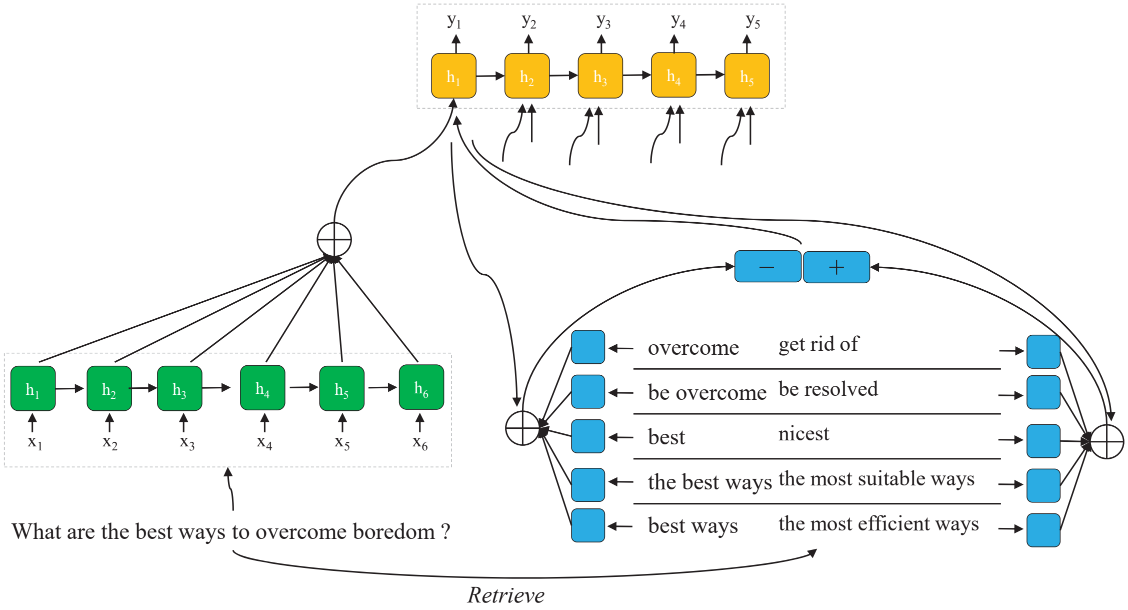

Figure 2: Architecture of dictionary-guided editing networks. At each step of the decoder, we implement the soft attention mechanism to guide the decision for word deletion or insertion.

We build our model on sequence-to-sequence with attention model, where the original sentence is the input sentence and the paraphrase sentence is the output sentence. We leverage the soft attention mechanism to encode the retrieved paraphrase pairs into edit vector. At each step of the decoder, our model concatenate edit vector to the input of the decoder.

For original sentence X, we first regard the output of the BiRNN as the representation of the original sentence X and use the standard attention model (Luong, Pham, and Manning 2015) to capture original-side information.

For the paraphrase pairs $\mathcal{D}$, we first implement a neutral encoder to convert it into a set of vectors and then use the soft attention to encode an edit vector.

In the case of single word paraphrase pairs, a good representation vector can be the word embedding of o_i or p_i. For multiple words, phrase o_i or phrase p_i can be represented as the sum of the individual word vectors (Gupta et al. 2018) as follows:

$$o_r^i = \sum_{w \in o_i} \Phi(w) \tag{2}$$

$$p_r^i = \sum_{w \in p_i} \Phi(w) \tag{3}$$

where $\Phi(w)$ is the word vector for word w and o_r^i is the representation vector of phrase o_i and p_r^i is the representation vector of phrase p_i. For each paraphrase pair in $\mathcal{D}$, we employ the same encoding method and convert the paraphrase pair $\mathcal{D}$ into representation vectors $\mathcal{D}' = \{(o_r^i, p_r^i)\}_{i=1}^M$, which contains $2 \times M$ vectors.

For representation vectors $\mathcal{D}'$, we adopt the soft attention mechanism, which is introduced to better utilize paraphrase pairs information. The soft attention mechanism can be used to guide the decision for word deletion or insertion in each step of the decoder. For instance in Figure 1, our dictionary-guided edit model pays more attention on the pair ("*overcome*", "*get rid of*") and pair ("*be overcome*", "*be resolved*") when it revises the word "*overcome*".

For the t-th time step, $\mathbf{h_t}$ denotes its hidden state of the decoder. $\mathbf{h_t}$ is computed via:

$$\mathbf{h_t} = f(\mathbf{h_{t-1}}, \mathbf{y_{t-1}}) \tag{4}$$

where function f is the gated recurrent unit (GRU) (Chung et al. 2014).

We compute a context vector $\mathbf{c_t}$ with hidden state $\mathbf{h_t}$ and paraphrase pairs as input to captures paraphrase pairs side information to guide the decoder. In paraphrase pairs $\mathcal{D} = \{(o_i, p_i)\}_{i=1}^M$, o_i might be the word that will be deleted and p_i might be inserted. In order to better guide our model on which word might be deleted or be inserted, we employ two soft attentions to compute the o_i-side and p_i-side context vectors respectively. Context vector $\mathbf{c_t}$ is computed as the weighted sum of o_r^i and p_r^i as follows:

$$\mathbf{c_t} = \sum_{i=1}^{M} \mathbf{a_{t,i}} \cdot o_r^i \oplus \sum_{i=1}^{M} \mathbf{a'_{t,i}} \cdot p_r^i \tag{5}$$

where the $\mathbf{a_{t,i}}$ is o_r^i-side alignment vector and $\mathbf{a'_{t,i}}$ is p_r^i-side alignment vector, whose size both equals M. The alignment vector $\mathbf{a_{t,i}}$ is formulated as:

$$\mathbf{a_{t,i}} = \frac{exp(score(\mathbf{h_t}, o_r^i))}{\sum_{j=1}^{M} exp(score(\mathbf{h_t}, o_r^i))} \tag{6}$$

$$score(\mathbf{h_t}, o_r^i) = \mathbf{v}^\top tanh(\mathbf{W}_\alpha[\mathbf{h_t} \oplus o_r^i]) \tag{7}$$

where $\mathbf{W}_\alpha$ and $\mathbf{v}$ are parameters. The p_r-side alignment vector $\mathbf{a'_{t,i}}$ is formulated as:

$$\mathbf{a'_{t,i}} = \frac{exp(score(\mathbf{h_t}, p_r^i))}{\sum_{j=1}^{M} exp(score(\mathbf{h_t}, p_r^i))} \quad (8)$$

$$score(\mathbf{h_t}, p_r^i) = \mathbf{v'}^\top tanh(\mathbf{W}'_\alpha[\mathbf{h_t} \oplus p_r^i]) \quad (9)$$

where $\mathbf{W}'_\alpha$ and $\mathbf{v}'$ are the attention parameters. We can observe values of alignment vectors $\mathbf{a_{t,i}}$ and $\mathbf{a'_{t,i}}$ to learn our dictionary how to guide the decoder on which word might be deleted or be inserted.

A softmax layer is introduced to compute probability distribution of the t-th time word:

$$\mathbf{y_t} = softmax(\mathbf{W_y}[\mathbf{y_{t-1}} \oplus \mathbf{h_t} \oplus \mathbf{c_t} \oplus \mathbf{c'_t}] + \mathbf{b_y}) \quad (10)$$

where $\mathbf{W_y}$ and $\mathbf{b_y}$ are both parameters. $\mathbf{c'_t}$ is computed as the weighted sum of the original hidden states (Luong, Pham, and Manning 2015).

For the generative model, the learning goal is to maximize the probability of the actual paraphrase $\mathbf{y}^*$. We learn our model by minimizing the negative log-likelihood (NLL):

$$\mathcal{J} = -\log(p(\mathbf{y}^*|\mathbf{x}, \mathcal{D}')) \quad (11)$$

The mini-batched Adam (Kingma and Ba 2014) algorithm is used to optimize the objective function. In order to avoid overfitting, we adopt dropout layers between different GRU layers same as (Zaremba, Sutskever, and Vinyals 2014).

Experiments

Datasets

We present the performance of our model on two benchmark paraphrase generation datasets, namely the MSCOCO and Quora datasets.

MSCOCO (Lin et al. 2014) is a large-scale captioning dataset which contains human annotated captions of over 120K images [2]. This dataset was used previously to evaluate paraphrase generation methods (Prakash et al. 2016; Gupta et al. 2018). In the MSCOCO dataset, each image has five captions from five different annotators. Annotators describe the most obvious object or action in an image, which makes this dataset very suitable for the paraphrase generation task. This dataset comes with separate subsets for training and validation: *Train 2014* contains over 82K images and *Val 2014* contains over 40K images. From the five captions accompanying each image, we randomly omit one caption and use the other four as training instances to create paraphrase pairs. In order to compare our results with previous work (Prakash et al. 2016; Gupta et al. 2018), 20K instances are randomly selected from the data for testing, 10K instances for validation and remaining data over 320K instances for training.

Quora dataset is related to the problem of identifying duplicate questions[3]. It consists of over 400K potential question duplicate pairs. The non-duplicate pairs are related questions or have similar topics, which are not truly semantically equivalent, so we use true examples of duplicate pairs as paraphrase generation dataset. There are a total of 150K such questions. 140K instances are randomly selected for training, 5K instances for validation and about 5K instances for testing.

Evaluation Metric

Automatic Evaluation Metric To automatically evaluate the performance of paraphrase generation models, we use the well-known evaluation metrics[4] for comparing parallel corpora: BLEU (Papineni et al. 2002) and METEOR (Lavie and Agarwal 2007). Previous work has shown that these metrics can perform well for paraphrase detection (Madnani, Tetreault, and Chodorow 2012) and correlate well with human judgments in paraphrase generation (Wubben, Van Den Bosch, and Krahmer 2010).

BLEU considers exact matching between reference paraphrases and system generated paraphrases by considering n-gram overlaps. METEOR uses stemming and synonymy in WordNet to improve and smoothen this measure. We report our p-values at 95% Confidence Intervals (CI).

Human Evaluation Metric In addition to the automatic metrics, we also ask human annotators to judge the quality of the generated paraphrases. We randomly sample 500 input sentences from both MSCOCO and Quora dataset, and ask three human evaluators to annotate each generated paraphrase. Following Gupta et al., generated paraphrases are annotated from two aspects *Relevance* (the paraphrase sentence is semantically close to the original sentence) and *Readability* (the paraphrase sentence is fluent and grammatically correct). We use 3-scale rating: +2, +1 and 0 for both aspects, where 0 is worst and 2 is best.

Implementation Details

We leverage the PPDB to build our paraphrased dictionary index and we have introduced our retrieval strategy before. The Paraphrased Database (PPBD)[5] is used to divide the database into six sizes, from *S* up to *XXXL*. We build our paraphrased dictionary index using *L* size PPBD. PPDB contains five types of entailment relations and we exact paraphrase pairs with equivalent entailment relations to ensure the quality of our paraphrased dictionary.

We use NLTK (Bird and Loper 2004) to tokenize the sentences and keep words that appear more than 10 times in our vocabulary. Following the data preprocessing method in previous work (Prakash et al. 2016; Gupta et al. 2018), we reduce those captions to the size of 15 words (by removing the words beyond the first 15) for the MSCOCO dataset. The max length of phrases in PPDB is set to 7 and the size M of the retrieved paraphrase pairs is 10.

The training hyper-parameters are selected based on the results of the validation set. The dimensions of word embeddings is set to 300 and hidden vectors are set to 512 in

[2] http://cocodataset.org/

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

[4] We used the evaluation software available at https://github.com/jhclark/multeval

[5] http://paraphrase.org

Table 1: Results on MSCOCO dataset. Higher BLEU and METEOR score is better. Scores of the methods marked with * are taken from (Gupta et al. 2018).

Model	Beam size	BLEU	METEOR
Seq2Seq	1	29.9	24.7
Residual LSTM	1	34.6	26.1
VAE-SVG*	1	39.2	29.2
VAE-SVG-eq*	1	37.3	28.5
Our method	1	**40.5**	**30.3**
Seq2Seq*	10	33.4	25.2
Residual LSTM*	10	37.0	27.0
VAE-SVG*	10	41.3	30.9
VAE-SVG-eq*	10	39.6	30.2
Our method	10	**42.8**	**31.4**

Table 2: Results on Quora dataset. Higher BLEU and METEOR score is better. Scores of the methods marked with * are taken from (Gupta et al. 2018).

Model	Beam size	BLEU	METEOR
Seq2Seq	1	25.9	25.8
Residual LSTM	1	26.3	26.2
VAE-SVG*	1	25.0	25.1
VAE-SVG-eq*	1	26.2	25.7
Our method	1	**27.6**	**29.9**
Seq2Seq	10	27.9	29.3
Residual LSTM	10	27.4	28.9
VAE-SVG-eq*	10	**37.1**	**32.0**
Our method	10	28.4	30.6

Table 3: Human evaluation results on MSCOCO dataset. Fleiss Kappa is denoted as κ

Dataset	Input	Relevance/κ	Readability/κ
MSCOCO	Ground Truth	1.07 / 0.62	1.86 / 0.83
	Our Method	0.95 / 0.66	1.79 / 0.79
Quora	Ground Truth	1.56 / 0.69	1.81 / 0.80
	Our Method	1.52 / 0.67	1.79 / 0.77

the sequence encoder and decoder. The dimensions of the attention vector are also set to 512 and the dropout rate is set to 0.5 for regularization. The mini-batched Adam (Kingma and Ba 2014) algorithm is used to optimize the objective function. The batch size and base learning rates are set to 64 and 0.0001, respectively.

Baselines

We compare our method with the following baseline methods for paraphrase generation:

Seq2Seq: We implement the standard sequence to sequence with attention model (Bahdanau, Cho, and Bengio 2015), which is implemented in OpenNMT (Klein et al. 2017). All the settings are the same as our system.

Residual LSTM: Residual LSTM is a stacked residual LSTM network under the sequence to sequence framework proposed by (Prakash et al. 2016). It adds residual connections between LSTM layers to help retain essential words in the generated paraphrases. We use the released code to conduct our experiments (https://github.com/iamaaditya/neural-paraphrase-generation).

VAE-SVG: VAE-SVG is the current state-of-the-art paraphrase generation method on the MSCOCO dataset (Gupta et al. 2018). It combines the variational autoencoder (VAE) and sequence-to-sequence model by conditioning the encoder and decoder sides of the VAE on the input sentence to generate paraphrases.

VAE-SVG-eq: VAE-SVG-eq is the current state-of-the-art paraphrase generation method on the Quora dataset (Gupta et al. 2018). Different from the VAE-SVG model, it makes the encoder of the original sentence same on both sides i.e. encoder side and the decoder side in this variation, which reduces the number of model parameters.

Evaluation Results

In Table 1 and 2, we present the results from various models for the MSCOCO and Quora datasets respectively. In our experiments, we compare our model on greedy search (beam size as 1) and beam search (beam size as 10).

As shown in Table 1, we compare our dictionary-guided editing networks model with several state-of-the-art methods on the MSCOCO dataset. The results demonstrate that our model consistently improves performance over other models for both greedy search and beam search. For greedy search, we are able to achieve more than 1.3 than the state-of-the-art in BLEU score, and 1.1 boost in METEOR and for beam search 1.5 boost in BLEU score and 0.5 performance improvement in METEOR score. For MSCOCO, the comparison between two models is significant at 95% CI, if the difference in their score is more than 0.2 in BLEU and 0.1 in METEOR.

In Table 2, we report BLEU and METEOR results for the Quora dataset. The results demonstrate that our proposed model outperforms other models at the non-beam search. Comparison between two models is significant at 95% CI, if the difference in their score is more than 0.2 in BLEU and 0.1 in METEOR for Quora dataset. For the greedy search, our model is able to give a 1.3 performance improvement for BELU and 3.7 improvement for the METEOR metric over the state-of-the-art one. For beam size of 10, our model outperforms other models in the Quora dataset except the VAE-SVG-eq model, in which beam search gives an 11 absolute point performance improvement in BLEU score and 2.6 improvement in METEOR score. However, beam search does not give such a significant improvement in our model, which we will discuss later in the following section.

Human evaluation results on both MSCOCO and Quora dataset are shown on Table 3, consisting of results from our method and ground truth and the agreements among the three labelers for each model. We compute the agreement for each input by Fleiss' kappa (Fleiss 1971). The values of the agreement are larger than 0.6 for both models and metrics, which are considered as "moderate agreement". From Table 3, we can see that our generated paraphrases are very close to the ground truth for both metrics *Relevance* and *Readability*. In particularly, for Quora dataset, the gaps be-

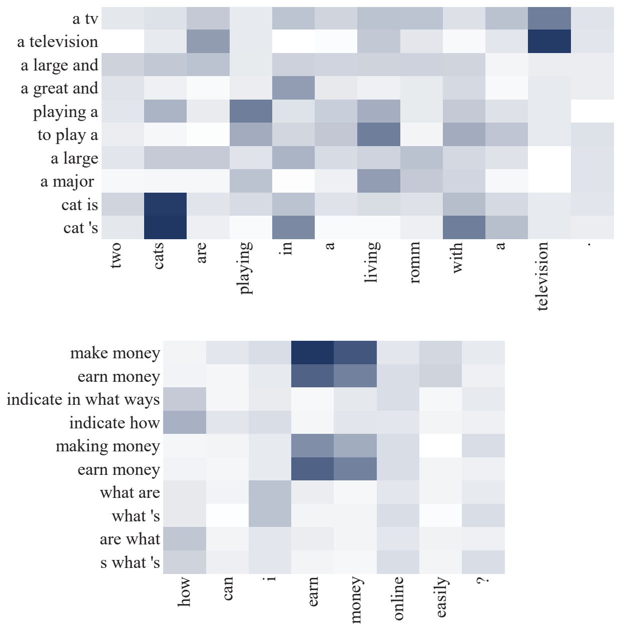

Figure 3: Visualization of dictionary-guided attention in the decoder. Each column in the diagram corresponds to the weights of the decoder and items in the paraphrase pairs.

tween our model and the ground truth are 0.04 in *Relevance* and 0.02 in *Readability*.

Furthermore, we note that the *Relevance* scores of MSCOCO dataset are lower than a perfect score of 2. Because MSCOCO dataset is an image caption dataset which allows annotators to describe things in an image in a considerable variation.

Discussions

Dictionary-Guided Attention In Figure 3, we show the visualization of dictionary-guided attention in the decoder. Each column in the diagram corresponds to the weights of the decoder and items in the paraphrase pairs.

Figure 3 shows two examples separately from MSCOCO and Quora datasets. Each example has five paraphrase pairs. The delete attention and insert attention scores are represented by gray scales and are column-wisely normalized as described in Equation 6 and 8. As described, the editing attention mechanism learns soft alignment scores between paraphrased dictionary and generated words. These scores are used to guide the decision for word deletion or insertion in the decoder.

In the first example, the generated paraphrase is *"two cats are playing in a living room with a television ."*. We find that the pair (*"a tv"*, *"a television"*) has larger attention scores where the decoder generates the word *television*. We also observe that when the decoder generates the word *cats*, the pair (*"cat is"*, *"cat 's"*) has a larger attention weight than other pairs. However, our model doesn't insert the phrase *cat 's* into generated sentence, which means that our dictionary-guided editing networks ensure the new sentence is grammatically correct during editing. For the second example in the Quora dataset, the model learns alignments pairs (*"make money"*, *"earn money"*) and (*"making money"*, *"earn money"*) when the decoder generates *earn money*. These examples demonstrate our paraphrase pairs have more effect on generating some words which might be deleted or inserted.

Table 4: Beam search improvement results on MSCOCO and Quora datasets.

Dataset	Model	Beam=1	ΔBLEU
MSCOCO	Seq2Seq	29.9	3.5
	Residual LSTM	34.6	2.4
	VAE-SVG-eq	37.3	2.3
	Our method	40.5	2.3
Quora	Seq2Seq	25.9	2.0
	Residual LSTM	26.3	1.1
	VAE-SVG-eq	26.2	10.9
	Our method	27.6	0.8

Beam Search Improvement In this section, we study the impact of beam search to these methods. As shown in Table 4, we list beam search improvement results on MSCOCO and Quora datasets. For the MSCOCO dataset, our beam search improvements are comparable to other models. For the Quora dataset, our BLEU boost is also comparable to Seq2seq model and Residual LSTM model, except the VAE-SVG-eq model which achieves surprise improvements from 26.2 to 37.1 through beam search. From results on MSCOCO dataset, we note that when the results of the greedy search become better, performance improvements through beam search get smaller for these methods. For Quora dataset, we also observe the similar trend for these methods except the VAE-SVG-eq model.

Case Study In Table 5, we show some generated paraphrase examples on MSCOCO and Quora datasets. In these tables, we highlight these paraphrase pairs which might be used to guide paraphrase generation and phrase pairs which are found in the retrieved paraphrase pairs in bold.

In Table 5, we can find there exist revised pairs that come from the retrieved paraphrase pairs. For example, for the MSCOCO dataset, for the source sentence *these two cats are playing in a room that has a large tv and a laptop computer .*, its paraphrase pairs contain a pair (*"a tv"*, *"a television"*). Our model replaces the phrase *a large tv* with *a television* and generates its paraphrase *two cats are playing in a living room with a television .* For the Quora dataset, for the input question *can you offer me any advice on how to lose weight ?*, our model generates *can you give me some advice on losing weight ?* as its paraphrase. We note that the editing model inserts the phrase *give me some advice* into the generated question and deletes the phrase *offer me any advice* from the source question. The retrieved paraphrase pairs contain a pair (*"offer advice"*, *"give advice"*), which is not the exact same as the insertion phrase and deletion phrase. It indicates our model can learn some soft operations from the dictionary and make small changes over the

Table 5: Example paraphrases generated using the dictionary-guided editing networks on MSCOCO and Quora datasets.

Source	these two cats are playing in a room that has **a large tv** and a laptop computer .
Reference	a cat being lazy and a cat being nozy in a living room with tv and a laptop displaying the same things .
Generated	two cats are playing in a living room with **a television** .
Dictionary	**(a tv, a television)** (a large and, a great and) (playing a, to play a)
Source	a large passenger **airplane** flying through the air .
Reference	an airplane that is , either , landing or just taking off .
Generated	a large **jetliner** flying through a blue sky .
Dictionary	(the airplane, the aeroplane) **(airplane, jetliner)** (a large, a great)
Source	what are ways i can **make money** online ?
Reference	can i earn money online ?
Generated	how can i **earn money** online easily ?
Dictionary	**(make money, earn money)** (indicate in what ways, indicate how) **(making money, earn money)**
Source	can you **offer** me any advice on how to lose weight ?
Reference	how can i efficiently lose weight ?
Generated	can you **give** me some advice on losing weight ?
Dictionary	(offer advice, provide advice) **(offer advice, give advice)** (you lost weight, you 've lost weight)

new sentence. As we can see, our model is able to replace some words or phrases in the original sentence based on the dictionary and makes necessary changes to ensure the new sentence is grammatically correct and fluent.

Related Work

Paraphrase generation aims to generate a semantically equivalent sentence with different expressions. Prior approaches can be categorized into knowledge-based approaches and statistical machine translation (SMT) based approaches. Knowledge-based approaches primarily rely on hand-crafted rules and dictionaries that enjoy high precision but that are hard to scale up. The pioneer of this approach is Kozlowski et al. (2003) who first pair simple semantic structures with their syntactic realization and then generate paraphrases using such predicate/argument structures. A famous paraphrase generation system is designed by Hassan et al. (2007), where paraphrases are generated by word substitutions and the substitution table is obtained by leveraging several external resources, such as WordNet and Microsoft Encarta encyclopedia. Subsequently, Madnani and Dorr (2010) propose a knowledge-driven method by using hand crafted rules or automatically learned complex paraphrase patterns (Zhao et al. 2009). SMT based paraphrase generation is proposed by (Quirk, Brockett, and Dolan 2004), where an SMT model is trained on large volumes of sentence pairs extracted from clustered news articles. Zhao et al. (2008) combine multiple resources to learn phrase-based paraphrase tables and corresponding feature functions to devise a log-linear SMT model. To leverage the power of multiple machine translate engine, a multi-pivot approach is proposed in (Zhao et al. 2010) to obtain plenty of paraphrase candidates. Then these candidates are used by selection-based and decoding-based methods to produce high-quality paraphrases.

Recently, deep learning-based approaches have been introduced for paraphrase generation and achieved great success. Prakash et al. (2016) employ the residual recurrent neural networks for paraphrase generation, that is one of the first major words that uses a deep learning model for this task. Gupta et al. (2018) propose a combination of variational autoencoder (VAE) and sequence-to-sequence model to generate paraphrase. We also investigate deep learning for paraphrase generation, and we are the first one to utilize an editing mechanism for this task.

Our work is in the spirit of prototype editing methods for natural language generation (Guu et al. 2017), which proposes a generative model that first samples a prototype sentence from training data and then edits it into a new sentence. We utilize the original sentence as a prototype and learn the edit vector from paraphrase dataset (PPDB) (Ganitkevitch, Van Durme, and Callison-Burch 2013). Li et al. (2018) introduce a simple approach for style transfer. It can be considered for applying content words by deleting phrases associated with original attribute values as a prototype, and combining a new phrase with the target attribute to generate a final output. Cao et al. (2018) employ existing summaries as soft templates, and rerank these soft templates by considering the current document. Finally, a summary is generated with a sequence-to-sequence framework augmented with the templates. Our work can be seen as an extension of editing methods for paraphrase generation. The stark difference is that our model is capable of leveraging an external dictionary in editing, which ensures that the expression changes do not affect its original semantic.

Conclusion

In this paper, we present a dictionary-guided editing networks model for generating paraphrase sentences through editing the original sentence. It can effectively leverage word level and phrase level paraphrase pairs from an off-the-shelf dictionary. The system jointly learns the selection of the appropriate word level and phrase level paraphrase pairs in the context of the original sentence from the Paraphrase Database (PPDB) as well as the generation of fluent natural language sentences. Experiments on the Quora and MSCOCO datasets demonstrate that the dictionary-guided editing networks significantly improves the existing generative models for paraphrase generation from scratch. The dictionary-guided editing networks can also be applied to other text generation tasks, such as the text style transfer where we can use word and phrase level style mapping dictionaries to facilitate sentence level style transfer results.

Acknowledgement

This research is supported by the National Natural Science Foundation of China (Grant No. 61732002).

References

Bahdanau, D.; Cho, K.; and Bengio, Y. 2015. Neural machine translation by jointly learning to align and translate. *ICLR*.

Bird, S., and Loper, E. 2004. Nltk: the natural language toolkit. In *Proceedings of the ACL 2004 on Interactive poster and demonstration sessions*, 31. Association for Computational Linguistics.

Cao, Z.; Li, W.; Wei, F.; and Li, S. 2018. Retrieve, rerank and rewrite: Soft template based neural summarization. Association for Computational Linguistics.

Chung, J.; Gulcehre, C.; Cho, K.; and Bengio, Y. 2014. Empirical evaluation of gated recurrent neural networks on sequence modeling. *NIPS 2014 Deep Learning and Representation Learning Workshop*.

Fleiss, J. L. 1971. Measuring nominal scale agreement among many raters. *Psychological bulletin* 76(5):378.

Ganitkevitch, J.; Van Durme, B.; and Callison-Burch, C. 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*, 758–764.

Gupta, A.; Agarwal, A.; Singh, P.; and Rai, P. 2018. A deep generative framework for paraphrase generation. *AAAI*.

Guu, K.; Hashimoto, T. B.; Oren, Y.; and Liang, P. 2017. Generating sentences by editing prototypes. *TACL*.

Hassan, S.; Csomai, A.; Banea, C.; Sinha, R.; and Mihalcea, R. 2007. Unt: Subfinder: Combining knowledge sources for automatic lexical substitution. In *Proceedings of the 4th International Workshop on Semantic Evaluations*, 410–413. Association for Computational Linguistics.

Kingma, D. P., and Ba, J. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Klein, G.; Kim, Y.; Deng, Y.; Senellart, J.; and Rush, A. M. 2017. Opennmt: Open-source toolkit for neural machine translation. In *Proc. ACL*.

Kozlowski, R.; McCoy, K. F.; and Vijay-Shanker, K. 2003. Generation of single-sentence paraphrases from predicate/argument structure using lexico-grammatical resources. In *Proceedings of the second international workshop on Paraphrasing-Volume 16*, 1–8. Association for Computational Linguistics.

Lavie, A., and Agarwal, A. 2007. Meteor: An automatic metric for mt evaluation with high levels of correlation with human judgments. In *Proceedings of the Second Workshop on Statistical Machine Translation*, 228–231. Association for Computational Linguistics.

Li, J.; Jia, R.; He, H.; and Liang, P. 2018. Delete, retrieve, generate: A simple approach to sentiment and style transfer. *NAACL*.

Lin, T.-Y.; Maire, M.; Belongie, S.; Hays, J.; Perona, P.; Ramanan, D.; Dollár, P.; and Zitnick, C. L. 2014. Microsoft coco: Common objects in context. In *European conference on computer vision*, 740–755. Springer.

Luong, M.-T.; Pham, H.; and Manning, C. D. 2015. Effective approaches to attention-based neural machine translation. *arXiv preprint arXiv:1508.04025*.

Madnani, N., and Dorr, B. J. 2010. Generating phrasal and sentential paraphrases: A survey of data-driven methods. *Computational Linguistics* 36(3):341–387.

Madnani, N.; Tetreault, J.; and Chodorow, M. 2012. Re-examining machine translation metrics for paraphrase identification. In *Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 182–190. Association for Computational Linguistics.

Papineni, K.; Roukos, S.; Ward, T.; and Zhu, W.-J. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting on association for computational linguistics*, 311–318. Association for Computational Linguistics.

Pavlick, E.; Rastogi, P.; Ganitkevitch, J.; Van Durme, B.; and Callison-Burch, C. 2015. Ppdb 2.0: Better paraphrase ranking, fine-grained entailment relations, word embeddings, and style classification. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, volume 2, 425–430.

Prakash, A.; Hasan, S. A.; Lee, K.; Datla, V.; Qadir, A.; Liu, J.; and Farri, O. 2016. Neural paraphrase generation with stacked residual lstm networks. *COLING*.

Quirk, C.; Brockett, C.; and Dolan, B. 2004. Monolingual machine translation for paraphrase generation. *EMNLP*.

Wubben, S.; Van Den Bosch, A.; and Krahmer, E. 2010. Paraphrase generation as monolingual translation: Data and evaluation. In *Proceedings of the 6th International Natural Language Generation Conference*, 203–207. Association for Computational Linguistics.

Zaremba, W.; Sutskever, I.; and Vinyals, O. 2014. Recurrent neural network regularization. *arXiv preprint arXiv:1409.2329*.

Zhao, S.; Niu, C.; Zhou, M.; Liu, T.; and Li, S. 2008. Combining multiple resources to improve smt-based paraphrasing model. *Proceedings of ACL-08: HLT* 1021–1029.

Zhao, S.; Lan, X.; Liu, T.; and Li, S. 2009. Application-driven statistical paraphrase generation. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP: Volume 2-Volume 2*, 834–842. Association for Computational Linguistics.

Zhao, S.; Wang, H.; Lan, X.; and Liu, T. 2010. Leveraging multiple mt engines for paraphrase generation. In *Proceedings of the 23rd International Conference on Computational Linguistics*, 1326–1334. Association for Computational Linguistics.