Paraphrase Generation: A Survey of the State of the Art

Jianing Zhou and **Suma Bhat**
University of Illinois at Urbana-Champaign
{zjn1746, spbhat2}@illinois.edu

Abstract

This paper focuses on paraphrase generation, which is a widely studied natural language generation task in NLP. With the development of neural models, paraphrase generation research has exhibited a gradual shift to neural methods in the recent years. This has provided architectures for contextualized representation of an input text and generating fluent, diverse and human-like paraphrases. This paper surveys various approaches to paraphrase generation with a main focus on neural methods.

1 Introduction

Paraphrases are texts that convey the same meaning while using different words or sentence structures. The generation of paraphrases is a longstanding problem for natural language learning. For example, the question *How do I improve my English* could be equivalently phrased as *What is the best way to learn English*. Paraphrasing can be play an important role in language understanding tasks, such as question answering (Dong et al., 2017; Zhu et al., 2017), machine translation (Seraj et al., 2015; Thompson and Post, 2020a), and semantic parsing (Berant and Liang, 2014; Cao et al., 2020). And it is also a good way for data augmentation (Kumar et al., 2019; Gao et al., 2020). Given a sentence, paraphrase generation aims to create its paraphrases that can have a different wording or different structure from the original sentence, while preserving the original meaning.

The focus of paraphrase generation has exhibited a gradual shift from classical approaches to more advanced neural approaches in the recent years with the rapid development of various neural models. Neural models have changed the traditional way paraphrase generation is performed and also provided new directions and architectures for the NLP community.

While several surveys on the traditional methods and limited neural methods for paraphrase generation have been published (Metzler et al., 2011; Gupta and Krzyżak, 2020), there is no thorough and comprehensive survey on neural methods for paraphrase generation. To our best knowledge, this is the first survey on neural methods for paraphrase generation. Therefore, our goal in this paper is to provide a timely survey on paraphrase generation, with a main focus on neural methods.

Sentences	Paraphrases
How do I improve my English	What is the best way to learn English
How far is Earth from Sun	What is the distance between Sun and Earth
if at any time in the preparation of this product the integrity of this container is compromised it should not be used .	this container should not be used if the product is compromised at any time in preparation .

Table 1: Examples of paraphrases from available datasets for paraphrase generation.

In the following section, we will first introduce the most frequently used datasets for paraphrase generation (Section 2). Then we list the traditional evaluation metrics in Section 3. In Section 4, we present some of the traditional approaches that were used before the neural methods. Neural models, the main focus of this paper, will be discussed in Section 5. After introducing all the methods, we compare the performance of the different models for paraphrase generation in Section 6. Finally, we identify some research gaps in paraphrase generation.

2 Datasets

In this section, we describe several datasets that have been extensively used for paraphrase generation.

PPDB The paraphrase database (Ganitkevitch et al., 2013) contains over 220 million paraphrase pairs, consisting of 73 million phrasal and 8 million lexical paraphrases, as well as 140 million para-

Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, pages 5075–5086
November 7–11, 2021.

Dataset	Parallel	Genre	Size	Gold Size	Length	Char Len
PPDB	✔	Phrase, words	220,000,000	220,000,000	2.85	16.25
WikiAnswer	✔	Question	18,000,000	18,000,000	11.43	54.33
MSCOCO	✔	Description	493,186	493,186	10.48	51.56
Quora	✔	Question	404,289	149,263	11.14	52.89
Twitter URL	✔	Twitter	2,869,657	116,000	14.80	-
ParaNMT	✔	Novels, laws	51,409,585	51,409,585	12.94	59.18

Table 2: Highlights of primarily used paraphrase generation datasets. Gold Size represents the size of the subset used for paraphrase generation when the original dataset was not used for generation. Length is the the average number of words per sentence and Char Len is the average number of characters per sentence.

phrase patterns that capture meaning-preserving syntactic transformations. Each paraphrase pair in PPDB contains a set of associated scores including paraphrase probabilities and monolingual distributional similarity scores. Despite its size and variety, because this dataset only contains phrasal and lexical paraphrases without any sentence paraphrases, it has recently fallen out of use.

WikiAnswer This dataset (Fader et al., 2013) contains approximately 18 million word-aligned question pairs that are paraphrases. The word alignments provided by this dataset also relate the synonyms in the paraphrase sentences. However, all the sentences provided in this dataset are questions, which restricts the paraphrases to only questions.

MSCOCO MSCOCO (Lin et al., 2014) was originally described as a large-scale object detection dataset. It contains human-annotated captions of over 120K images, and each image is associated with five captions from five different annotators. There are about 500K pairs of paraphrases in this dataset. In most cases, annotators describe the most prominent object/action in an image, which makes this dataset suitable for paraphrase-related tasks.

Quora Quora[1] released a dataset in 2017, which consists of over 400K lines of potential question duplicate pairs. Among these potential question duplicate pairs, there are 150K question pairs annotated as paraphrases. For the paraphrase generation task only use these valid paraphrase question pairs are used for training and testing. Like WikiAnswer, this dataset is restricted to questions.

Twitter URL Twitter URL (Lan et al., 2017) is constructed by collecting large-scale sentential paraphrases from Twitter by linking tweets through shared URLs. This dataset consists of two subsets, each of which contains both paraphrases and non-paraphrases. One subset is labeled by human annotators, and the other is labeled automatically. Only the paraphrase sentence pairs are used for paraphrase generation. Because this dataset includes sentence pairs that are labeled automatically (as paraphrase or not), the annotation is noisy.

ParaNMT ParaNMT (Wieting and Gimpel, 2018) is a dataset of more than 50 million English-English sentential paraphrase pairs. The pairs were generated automatically by using back-translation to translate the non-English side of a large Czech-English parallel corpus. Owing to its recency, it has not been used widely.

3 Evaluation Methods

Two general types of evaluation metrics are commonly used to evaluate paraphrase generation: automatic evaluation and human evaluation.

Automatic Evaluation Several automatic evaluation metrics are used for the evaluation of paraphrase generation. The widely-used metrics include (1) BLEU (Papineni et al., 2002), which was originally developed to evaluate machine translation systems; (2) METEOR (Denkowski and Lavie, 2014), which aims to address BLEU's weakness of being unable to measure semantic equivalents when applied to low-resource languages and has a better correlation with human judgment at the sentence/segment level than BLEU; (3) ROUGE (Lin, 2004), a recall-based evaluation metric originally developed for text summarization, has also been used to evaluate paraphrase generation. Its versions, ROUGE-N (computing the n-gram recall) and ROUGE-L (focusing on the longest common subsequence) are mostly used. (4) TER (Snover

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

et al., 2006), which was also developed to evaluate machine translation. It measures the number of edits that a human translator would have to perform to change a translation so it exactly matches a reference translation. A TER score is a value in the range of 0-1, but is frequently presented as a percentage, where lower is better.

Human Evaluation Due to the fact that automatic evaluation metrics mainly focus on the n-gram overlaps instead of meaning, human evaluation is used to provide a more accurate and qualitative evaluation of the generated output. In human evaluation, human annotators are asked to score generated paraphrases along multiple dimensions of quality such as similarity, clarity, and fluency. Owing to the manual annotation efforts, human evaluation is naturally more costly compared to automatic evaluation, but more representative of the quality of the generated output.

4 Traditional Approaches

In this section, some traditional approaches without neural models will be introduced.

Rule-Based Approaches

Rule-based paraphrase generation approaches build on hand-crafted or automatically collected paraphrase rules. In the early works, these rules were mainly hand-crafted (McKeown, 1983). Due to the significant manual efforts, some researchers have sought to collect paraphrase rules automatically (Lin and Pantel, 2001; Barzilay and Lee, 2003). However, the limitation of the extracting methods has led to the generation of long and complex paraphrase patterns, in turn impacting performance.

Thesaurus-Based Approaches

This approach usually generates paraphrases by substituting some words in the source sentences with their synonyms extracted from a thesaurus (Bolshakov and Gelbukh, 2004; Kauchak and Barzilay, 2006). Thesaurus-based approaches proceed by first extracting all synonyms from a thesaurus for the words to be replaced. Then the optimal candidate is selected according to the context in the source sentence. Although simple and effective, this approach is severely limited by the diversity of the generated paraphrases.

SMT-Based Approach

This approach is based on statistical machine translation (SMT) and is motivated by the fact that paraphrase generation can be seen as a special case of machine translation (i.e., monolingual machine translation). A machine translation model normally finds a best translation $\hat{e}$ of a text in language f to a text in language e by utilizing a statistical translation model $p(f|e)$ and a language model $p(e)$:

$$\hat{e} = arg\max_{e \in e^*} p(f|e)p(e)$$

Applying this idea to paraphrase generation, such a model will find a best paraphrase $\hat{t}$ of a text in the source side s to a text in the target side t obtained as,

$$\hat{t} = arg\max_{t \in t^*} p(s|t)p(t)$$

For instance, (Wubben et al., 2010) constructed a large-scale parallel corpus containing paraphrases collected from the headlines that appeared in Google News. Then they trained a Phrase-Based Machine Translation model (PBMT) (Koehn et al., 2007) on their parallel corpus using the MOSES package. The trained PBMT is finally used to generate paraphrases.

5 Neural Approaches

Early works on paraphrasing mainly focused on template-based or statistical machine translation approaches. However, the matching of templates and modeling of a statistical translation model are both challenging tasks. With the recent advances of neural networks, especially the sequence-to-sequence framework, Seq2Seq models were first use for paraphrase generation by (Prakash et al., 2016). Their work inspired the wide use of neural models for paraphrase generation. Below we introduce the main approaches based on neural models that are used for paraphrase generation.

5.1 Encoder-Decoder Architecture

Currently, most of the existing paraphrase generation models are based on sequence-to-sequence models consisting of an encoder and a decoder. The encoder will encode the source texts into a contextualized vector representation along with a list of vector representations capturing the semantics of each word and context. Then, the decoder will generate paraphrases based on the vectors given by the encoder.

Encoding Side

The main purpose of encoding is to extract the semantic information for the decoder to generate paraphrases. With the development of various neural models, researchers also have multiple choices for the encoder.

Encoder With a consistent goal of learning better abstract contextualized representation of the input text, several architectures have been explored by researchers. (Prakash et al., 2016) first utilized a seq2seq model implemented as recurrent neural networks—long short term memory networks (LSTMs) (Hochreiter and Schmidhuber, 1997)—to process long sequences. A nonvolutional neural network (CNN) has also been used to construct seq2seq models as a CNN has fewer parameters and thus is faster to train (Vizcarra and Ochoa-Luna, 2020). The Transformer model (Vaswani et al., 2017) has shown state-of-the-art performance on multiple text generation tasks. Due to the Transformer's improved ability to capture long-range dependencies in sentences, (Wang et al., 2019) utilized a Transformer to construct their seq2seq model. More recently, large language models using transformer architectures have achieved state-of-the-art results for many NLP tasks while using less supervised data than before. Therefore, some researchers also utilized large pretrained language models such as GPT-2 (Radford et al., 2019) and BART (Lewis et al., 2020) as their encoder-decoder framework (Witteveen and Andrews, 2019; Hegde and Patil, 2020; Garg et al., 2021).

Decoding Side

At the decoding side, the contextualized representation is used at each decoding step with the vector representation of previously generated words. Finally, a distribution over the vocabulary is obtained and the word with highest probability will be generated. This method is greedy decoding. Besides, a more commonly used method called *beam search* (Wiseman and Rush, 2016) is used, which identifies the k-best paths up to current timestep during decoding.

However, greedy decoding and beam search methods are both generic approaches for all text generation tasks without a specific focus on paraphrase generation. Therefore, with the goal of generating paraphrases and avoiding the words existing in the source sentences, a few blocking mechanisms have been proposed to prevent the decoder from generating the same words in the source sentences. This is also a way to guarantee the diversity of the generated paraphrases and prevent the models from directly copying the input into the output paraphrases (Niu et al., 2020; Thompson and Post, 2020b).

5.2 Improvements Based on Encoder-Decoder Architecture

The numerous attempts that have been made to improve the Encoder-Decoder architecture for paraphrase generation can be broadly categorized into two types based on their focus: **A. Model-focused**; and **B. Attribute-focused**. Next we introduce them respectively with more fine-grained divisions.

A. Model-focused

Model-focused improvements only aim to utilize various mechanisms to enhance the encoder or the decoder without paying special attention to the attributes of the generated paraphrases (e.g., granularity level such as word-level, phrase-level and sentence-level).

Attention The *Attention* mechanism (Bahdanau et al., 2015) enables the decoder to focus on some words/phrases that are of high relevance when generating a word. First, a weight for each token in the source sequence in each timestep is computed to indicate the importance, emphasizing the important information from the input and de-emphasizing the unimportant information. Given the weight distribution over all the tokens in the source sequence, this extra input vector, the context vector, is provided to the decoder..

Copy To counter the effect of rare and out-of-vocabulary words in neural sequence models, (Vinyals et al., 2015) proposed a pointer network. A pointer network copies an element from the input sequence directly into the output. Similarly, *copy* mechanism copies a span of elements from the input sequence decided by attention mechanism directly into the output. With *copy* mechanism, the decoder is able to determine whether a generate mode or a copy mode should be used at each timestep. First introduced by Gu et al. (2016) for abstractive summarization, Cao et al. (2017) haev also applied the *copy* mechanism to paraphrase generation. Despite the advantage of generating well-formed paraphrases by using the *copy* mechanism, it leads to the undesirable consequence of making a paraphrase contain many of the phrases

in the original sentence and limits diversity. This calls for a controlled use of the copy mechanism during paraphrase generation.

Variational autoencoder (VAE) The VAE (Kingma et al., 2014) is able to learn rich, nonlinear representations for high-dimensional inputs. Provided a latent representation $z \sim \mathcal{N}(\mu, \sigma)$ with the distribution learned from inputs by the encoder, the VAE decoder is equipped with the ability of producing realistic outputs conditioned on the latent representation and the learned distribution. The learning is achieved by reconstructing the original input from the latent code z. Therefore, with the help of VAE, the paraphrase patterns are encoded into the latent representation $z \sim \mathcal{N}(\mu, \sigma)$, which provides the model with control over the capacity of the learned distribution. Multiple paraphrase patterns and related words/phrases are grouped under the same latent assignment. Every time we sample a latent code z from the distribution $\mathcal{N}(\mu, \sigma)$, we get a new paraphrase pattern. Researchers have explored VAEs with different encoders and decoders. For examples, Gupta et al. (2017) implemented the encoder and decoder with LSTMs, whereas the transformer is utilized by Roy and Grangier (2019).

Reinforcement Learning As pointed out by (Ranzato et al., 2015), a well-known problem of the encoder-decoder architecture is *exposure bias*: the decoding of current word is conditioned on the *gold* references during training but on the generated output from the last timestep during testing. Therefore, the error might be accumulated and propagated when testing. Another problem lies in the mismatch between the training goal and the evaluation metrics. While the generated paraphrases are finally evaluated automatically using the previously mentioned metrics, the network is trained to maximize the probability of generating the reference paraphrases. Therefore, minimizing the training loss might not correspond to optimizing the evaluation metric. To address this limitation, reinforcement learning (RL) is leveraged. RL aims to train an agent to interact with the environment with the goal of maximizing its reward. Toward finding an optimal policy, RL can be used to maximize the reward indicated as a desired evaluation metric or a combination of multiple desired metrics. Rather than minimizing loss (the conventional approach), Li et al. (2018) first utilized RL to maximize the reward given by an evaluator which outputs a real value to represent the matching degree between two sentences as paraphrases of each other. Other reward functions have been explored by researchers, including ROUGE score, perplexity score and language fluency (Siddique et al., 2020; Liu et al., 2020).

Generative adversarial networks (GAN) Proposed by Goodfellow et al. (2014), GANs consist of generators and discriminators, where generators try to generate realistic outputs that match the real distribution and discriminators try to distinguish between the samples generated by generators and the samples that are real. GAN is originally trained by minimax optimization proposed in (Goodfellow et al., 2014). However, when GAN is applied in text generation, the traditional training method cannot be used because generating discrete words is non-differentiable. Therefore, the idea of policy gradient (Sutton et al., 1999) is leveraged to solve this problem (Yu et al., 2017). With policy gradient applied, discriminators act like the reward function in RL. Moreover, different discriminators can provide different desired rewards and thus equip the model with the capacity to generating text with different conditions. Here, a model is usually trained in an adversarial way: generators and discriminators are first pretrained, then generators are trained to maximize the loss of the fixed discriminators, then generators are fixed and discriminators are again trained to minimize the loss by provided the real samples and the samples generated by the fixed generators. For the task of paraphrase generation, different discriminators are designed to distinguish between generated samples and real samples, paraphrases and non-paraphrases (Yang et al., 2019; Vizcarra and Ochoa-Luna, 2020).

B. Attribute-focused

For attribute-focused improvements, their purpose is to improve the quality of generated paraphrases in some specific aspects such as diversity and also provide control over some attributes of generated paraphrases such as syntax and granularity level. These attribute-focused works usually use the previously mentioned models as their backbone models. Based on the backbone models, different mechanisms are applied for different focuses.

Diversity Attempts focusing on diversity aim to generate multiple diverse paraphrases for a given sentence. Some works control diversity by provid-

Models	Quora				MSCOCO			
	ROUGE-1	ROUGE-2	BLEU-2	BLEU-4	ROUGE-1	ROUGE-2	BLEU-2	BLEU-4
Seq2Seq (Prakash et al., 2016)	57.27	33.04	40.41	24.97	40.11	14.31	47.14	21.65
Seq2Seq-attn (Prakash et al., 2016)	57.10	32.86	40.49	24.89	41.07	15.26	49.65	23.66
Seq2Seq-attn-copy (Gu et al., 2016)	61.96	36.07	-	-	-	-	-	-
Seq2Seq-VAE (Gupta et al., 2017)	56.44	30.12	36.89	23.06	40.10	15.18	52.42	25.99
Transformer (Vaswani et al., 2017)	61.25	34.23	42.91	30.38	-	-	-	-
Seq2Seq-LBOW (Fu et al., 2019)	58.79	34.57	42.03	26.17	**42.08**	**16.13**	51.14	25.27
RbM (Li et al., 2018)	64.39	38.11	43.54	-	-	-	-	-
DB (Niu et al., 2020)	**67.49**	42.33	-	-	-	-	-	-
DNPG (Li et al., 2019)	63.73	37.75	-	25.03	-	-	-	-
FSET (Kazemnejad et al., 2020)	66.17	39.55	**51.03**	**33.46**	-	-	-	-
SCSVED (Chen et al., 2020)	60.28	35.26	41.56	27.37	40.90	15.70	**54.35**	**28.24**
SGCP (Kumar et al., 2020)	66.9	**45.0**	-	-	-	-	-	-

Table 3: State-of-the-art performance on Quora dataset and MSCOCO dataset.

ing control signals to the decoder. Random pattern embeddings are used by (Xu et al., 2018). (Kumar et al., 2019) utilized a submodular mechanism to maximize submodular functions measuring fidelity and diversity. (An and Liu, 2019), (Chen et al., 2020) and (Cao and Wan, 2020) all generate diverse paraphrases by providing the decoder with different latent patterns as control signal. Furthermore, (Cao and Wan, 2020) also incorporated their model with a diversity loss to control diversity. (Liu et al., 2020) use RL with multiple reward functions to generate diverse paraphrases. One of the reward functions computes ROUGE score between a generated sentence and original sentence, which can focus on the word variations and diversity. Acting like a reward function in RL, discriminators naturally can be used to provide control over some desired attributes. (Qian et al., 2019) utilized multiple generators in GAN to generate multiple diverse paraphrases. A generator discriminator is used to distinguish sentences generated by different generators and guarantee the generated paraphrases are diverse enough.

Word-Level Works on word-level paraphrasing mainly focus on generating paraphrases by replacing original words in the source texts with synonyms. Some works leveraged external linguistic knowledge (Cao et al., 2017; Lin et al., 2020). (Cao et al., 2017) utilized an alignment table capturing many synonym mappings based on the IBM Model (Chahuneau et al., 2013). (Lin et al., 2020) utilized WordNet (Miller, 1995) to retrieve synonyms. Other works instead proposed special mechanisms to learn a mapping of synonyms (Ma et al., 2018; Fu et al., 2019). For example, (Ma et al., 2018) utilized retrieved-based method to learn such a mapping. (Fu et al., 2019) incorporates a novel latent bag-of-word mechanism into seq2seq model for content planning, which mainly provides candidate synonyms for words in the source texts. However, generating paraphrases only on a word-level makes the quality and diversity of generated paraphrases limited. Therefore, paraphrasing has also been studied on other granularity level, e.g. syntax level.

Syntax Works in this category explore methods to provide control over the syntax of generated paraphrases. Basically, all the methods used by previous works can be split into two classes: 1. **Explicit Control** and 2. **Implicit Control**. Methods in the first class first encode the syntax tree of an exemplar sentence into a list of vector representations and then feed them into decoder at each timestep when decoding (Iyyer et al., 2018; Chen et al., 2019; Goyal and Durrett, 2020; Kumar et al., 2020). These methods can provide explicit control over the syntax of generated paraphrases and thus has better interpretability. The second class of methods will first learn a distribution over syntax information by VAE. Then a latent syntax variable sampled from the learned distribution will be fed into decoder at each decoding step (Chen et al., 2020). Although the control provided by this method is implicit, it does not require exemplar sentences and also can group multiple related syntax under the same latent assignment.

Multi-Level Focusing on a single granularity level of paraphrasing still makes generated paraphrases limited. Therefore, researchers also explore methods to combine multiple granularity levels together. Such attempts equip their model with the capacity of generating synonyms, substituting phrases and also rearrange sentential structures (Li et al., 2019; Huang et al., 2019; Kazemnejad et al.,

Models	Dataset					Evaluation				
	WikiAnswer	MSCOCO	Quora	Twitter	ParaNMT	BLEU	METEOR	ROUGE	TER	Human
(Prakash et al., 2016)	✔	✔	✗	✗	✗	✔	✔	✗	✔	✗
(Gupta et al., 2017)	✗	✔	✔	✗	✗	✔	✔	✗	✔	✔
(Fu et al., 2019)	✗	✔	✔	✗	✗	✔	✗	✔	✗	✗
(Li et al., 2018)	✗	✗	✔	✔	✗	✔	✔	✔	✗	✔
(Niu et al., 2020)	✗	✗	✔	✗	✔	✔	✗	✔	✗	✔
(Li et al., 2019)	✔	✗	✔	✗	✗	✔	✗	✔	✗	✔
(Kazemnejad et al., 2020)	✗	✗	✔	✔	✗	✔	✔	✔	✗	✔
(Chen et al., 2020)	✗	✔	✔	✗	✗	✔	✗	✔	✗	✗
(Kumar et al., 2020)	✗	✗	✔	✗	✔	✔	✔	✔	✗	✔

Table 4: Datasets and evaluation metrics used by different works on paraphrase generation. Twitter represents Twitter URL corpus. Only (Prakash et al., 2016) used PPDB. Therefore, we did not include PPDB into this table.

2020). By using multiple encoders, (Li et al., 2019) and (Kazemnejad et al., 2020) both enable their models to capture paraphrasing patterns on different granularity levels. (Huang et al., 2019) instead utilized the help of external linguistic knowledge from the paraphrase database (Ganitkevitch et al., 2013) to retrieve and learn word-level and phrase-level paraphrases. With different methods, both of them successfully combine multiple granularity levels together when generating paraphrases.

6 State-of-the-Art Performance

Table 3 shows the ROUGE and BLEU scores of state-of-the-art performance on some most frequently used evaluation corpus in recent years: MSCOCO and Quora. Due to the facts that different metrics are used in different works, different datasets are used in different works and many of them did not release their codes, Table 3 is not fully filled. However, with most of the table filled, we can still have some observations worth mentioning.

First, the use of attention mechanism achieves a close performance on Quora but has a better performance on MSCOCO (row 1 and 2). Similarly, the simple application of VAE also achieves a close performance on Quora but further improves the performance on MSCOCO (row4). With the *copy* mechanism, the Seq2Seq model is able to retain some words and thus yields a much better results (row 3). Transformer (row 5) outperforms all the Seq2Seq-based models without *copy* mechanism (row 1,2,4,6), which shows the advantages of Transformer and meanwhile also proves the effectiveness of *copy* mechanism.

Second, a model that employs RL (row 7) has a great advantage for generating better paraphrases because of the reward provided. Therefore, a well designed optimization goal plays an important role in the task of paraphrase generation.

Third, a novel decoding algorithm based on large pretrained language models helps to generate better paraphrases at the word level (row 8) because of the strength of large pretrained language models and the synonyms learned by decoding algorithm.

Fourth, the attempts to improve paraphrase generation with a special focus on combining multiple granularity levels also yield good performance (row 9,10). When learning to generate paraphrase in word level, phrase level and sentence level at the same time, their models improve the performance on multiple metrics compared with their backbone Transformer model (row 5).

Finally, incorporating syntax control into paraphrase generation will also yield better results at word level and sentence level (row 11,12). Compared with implicit control (row 11), explicit control has a much better performance (row 12) based on Quora.

It should be noted that most of the works utilize two datasets for experiments (as shown in Table 4) with one of them focusing on question paraphrases and the other focusing on general sentence paraphrases. Quora is the most popular dataset for question paraphrases. However, for corpus focusing on general sentnece paraphrases, different works have different choices among MSCOCO, Twitter URL and ParaNMT. MSCOCO is more preferred for less noise compared with Twitter URL and ParaNMT. Therefore, a combination of MSCOCO and Quora is more reasonable.

For evaluation metrics, BLEU is the most frequently used one. However, as proposed by (Niu et al., 2020), current automatic evaluation metrics are limited for evaluating paraphrase generation

BLEU	Quality	Target	Generated Paraphrase
High	High	a picture of someone taking a picture of herself	a woman taking a picture with a cell phone.
High	Low	a batter swinging a bat at a baseball	a batter swinging a baseball at a bat
Low	High	a man in sunglasses laying on a green kayak.	the man lying on a boat in the water.
Low	Low	people on a gold course enjoy a few games	a group of peaple walking

Table 5: Samples of generated paraphrases and their quality. Selected from (Niu et al., 2020).

because of "curse of BLEU on paraphrase evaluation". As shown in Table 5, examples with low BLEU scores might include both relatively good and bad paraphrasing because BLEU scores only measure the overlap between outputs and references. However, a generated paraphrase might still be a good paraphrase even it is not same with the reference. Therefore, for evaluation, it is better to combine automatic evaluation metrics and human evaluation together for a more comprehensive evaluation.

7 Conclusion

Although recent neural models have shown great advances, state-of-the-art results are still not satisfactory enough. Therefore, more advanced paraphrasing models still need to be explored. Below we discuss several potential directions of research that we believe are worth studying.

Pretrained language models Virtually all recent work related to the application of pretrained language models on paraphrase generation is quite naive. Therefore, we could combine the large pretrained language models with other mechanisms, for example reinforcement learning, VAE and GAN.

Multi-level controllable paraphrase generation Most recent works on multi-level paraphrase generation only focus on word-level paraphrasing and phrase-level paraphrasing. However, more granularity levels can be incorporated. We believe it is worthwhile to study the combination of various levels, including word-level, phrase-level, syntax-level and sentence-level.

Transfer learning With the goal of generating different surfaces of given sentences while preserving the meaning, text summarization, text simplification and paraphrase generation are essentially similar. Therefore, one could utilize transfer learning of these three tasks to improve the performance.

Stylistic paraphrase generation Currently, word- and phrase-substitution in paraphrase generation cannot be carefully controlled. Therefore, it is hard to control the style of generated paraphrases. We believe it is worthwhile to explore methods of incorporating specific styles into generated paraphrases. For instance, by controlling the types of words and phrases, we can incorporate metaphor and idiomatic expressions into paraphrases (Zhou et al., 2021b,a), which could also help to enhance creativity and diversity of generated paraphrases.

Evaluation metrics As stated in Section 6, BLEU scores and other automatic evaluation metrics based on similar principle are not good enough to evaluate paraphrase generation. Thus there is a need for better automatic evaluation methods. One possible method is to utilize paraphrase identification in the automatic evaluation metrics to explicitly provide an evaluation of if the generated sentence and input sentence are paraphrases.

References

Zhecheng An and Sicong Liu. 2019. Towards diverse paraphrase generation using multi-class wasserstein gan. *arXiv preprint arXiv:1909.13827*.

Dzmitry Bahdanau, Kyung Hyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. In *3rd International Conference on Learning Representations, ICLR 2015*.

Regina Barzilay and Lillian Lee. 2003. Learning to paraphrase: An unsupervised approach using multiple-sequence alignment. In *Proceedings of the 2003 Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics*, pages 16–23.

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.

Igor A Bolshakov and Alexander Gelbukh. 2004. Synonymous paraphrasing using wordnet and internet. In *International Conference on Application of Natural Language to Information Systems*, pages 312–323. Springer.

Ruisheng Cao, Su Zhu, Chenyu Yang, Chen Liu, Rao Ma, Yanbin Zhao, Lu Chen, and Kai Yu. 2020. Unsupervised dual paraphrasing for two-stage semantic parsing. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6806–6817.

Yue Cao and Xiaojun Wan. 2020. Divgan: Towards diverse paraphrase generation via diversified generative adversarial network. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: Findings*, pages 2411–2421.

Ziqiang Cao, Chuwei Luo, Wenjie Li, and Sujian Li. 2017. Joint copying and restricted generation for paraphrase. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 31.

Victor Chahuneau, Eva Schlinger, Noah A Smith, and Chris Dyer. 2013. Translating into morphologically rich languages with synthetic phrases. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1677–1687.

Mingda Chen, Qingming Tang, Sam Wiseman, and Kevin Gimpel. 2019. Controllable paraphrase generation with a syntactic exemplar. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5972–5984.

Wenqing Chen, Jidong Tian, Liqiang Xiao, Hao He, and Yaohui Jin. 2020. A semantically consistent and syntactically variational encoder-decoder framework for paraphrase generation. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 1186–1198.

Michael Denkowski and Alon Lavie. 2014. Meteor universal: Language specific translation evaluation for any target language. In *Proceedings of the ninth workshop on statistical machine translation*, pages 376–380.

Li Dong, Jonathan Mallinson, Siva Reddy, and Mirella Lapata. 2017. Learning to paraphrase for question answering. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 875–886.

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

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

Silin Gao, Yichi Zhang, Zhijian Ou, and Zhou Yu. 2020. Paraphrase augmented task-oriented dialog generation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 639–649.

Sonal Garg, Sumanth Prabhu, Hemant Misra, and G Srinivasaraghavan. 2021. Unsupervised contextual paraphrase generation using lexical control and reinforcement learning. *arXiv preprint arXiv:2103.12777*.

Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. 2014. Generative adversarial nets. In *Advances in Neural Information Processing Systems*, volume 27. Curran Associates, Inc.

Tanya Goyal and Greg Durrett. 2020. Neural syntactic preordering for controlled paraphrase generation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 238–252.

Jiatao Gu, Zhengdong Lu, Hang Li, and Victor OK Li. 2016. Incorporating copying mechanism in sequence-to-sequence learning. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1631–1640.

Ankush Gupta, Arvind Agarwal, Prawaan Singh, and Piyush Rai. 2017. A deep generative framework for paraphrase generation. *arXiv preprint arXiv:1709.05074*.

Varun Gupta and Adam Krzyżak. 2020. An empirical evaluation of attention and pointer networks for paraphrase generation. In *International Conference on Computational Science*, pages 399–413. Springer.

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

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation*, 9(8):1735–1780.

Shaohan Huang, Yu Wu, Furu Wei, and Zhongzhi Luan. 2019. Dictionary-guided editing networks for paraphrase generation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 6546–6553.

Mohit Iyyer, John Wieting, Kevin Gimpel, and Luke Zettlemoyer. 2018. Adversarial example generation with syntactically controlled paraphrase networks. 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 1875–1885.

David Kauchak and Regina Barzilay. 2006. Paraphrasing for automatic evaluation. In *Proceedings of the Human Language Technology Conference of the NAACL, Main Conference*, pages 455–462.

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.

Diederik P Kingma, Danilo J Rezende, Shakir Mohamed, and Max Welling. 2014. Semi-supervised learning with deep generative models. In *Proceedings of the 27th International Conference on Neural Information Processing Systems-Volume 2*, pages 3581–3589.

Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, et al. 2007. Moses: Open source toolkit for statistical machine translation. In *Proceedings of the 45th annual meeting of the association for computational linguistics companion volume proceedings of the demo and poster sessions*, pages 177–180.

Ashutosh Kumar, Kabir Ahuja, Raghuram Vadapalli, and Partha Talukdar. 2020. Syntax-guided controlled generation of paraphrases. *Transactions of the Association for Computational Linguistics*, 8:329–345.

Ashutosh Kumar, Satwik Bhattamishra, Manik Bhandari, and Partha Talukdar. 2019. Submodular optimization-based diverse paraphrasing and its effectiveness in data augmentation. 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 3609–3619.

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.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880.

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.

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.

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

Dekang Lin and Patrick Pantel. 2001. Discovery of inference rules for question-answering. *Natural Language Engineering*, 7(4):343–360.

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.

Zibo Lin, Ziran Li, Ning Ding, Hai-Tao Zheng, Ying Shen, Wei Wang, and Cong-Zhi Zhao. 2020. Integrating linguistic knowledge to sentence paraphrase generation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 8368–8375.

Mingtong Liu, Erguang Yang, Deyi Xiong, Yujie Zhang, Yao Meng, Changjian Hu, Jinan Xu, and Yufeng Chen. 2020. A learning-exploring method to generate diverse paraphrases with multi-objective deep reinforcement learning. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 2310–2321.

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.

Kathleen McKeown. 1983. Paraphrasing questions using given and new information. *American Journal of Computational Linguistics*, 9(1):1–10.

Donald Metzler, Eduard Hovy, and Chunliang Zhang. 2011. An empirical evaluation of data-driven paraphrase generation techniques. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 546–551.

George A Miller. 1995. Wordnet: a lexical database for english. *Communications of the ACM*, 38(11):39–41.

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

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

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

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

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.

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

Aurko Roy and David Grangier. 2019. Unsupervised paraphrasing without translation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 6033–6039.

Ramtin Mehdizadeh Seraj, Maryam Siahbani, and Anoop Sarkar. 2015. Improving statistical machine translation with a multilingual paraphrase database. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1379–1390.

AB Siddique, Samet Oymak, and Vagelis Hristidis. 2020. Unsupervised paraphrasing via deep reinforcement learning. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 1800–1809.

Matthew Snover, Bonnie Dorr, Richard Schwartz, Linnea Micciulla, and John Makhoul. 2006. A study of translation edit rate with targeted human annotation. In *Proceedings of association for machine translation in the Americas*, volume 200. Citeseer.

Richard S Sutton, David A McAllester, Satinder P Singh, Yishay Mansour, et al. 1999. Policy gradient methods for reinforcement learning with function approximation. In *NIPs*, volume 99, pages 1057–1063. Citeseer.

Brian Thompson and Matt Post. 2020a. Automatic machine translation evaluation in many languages via zero-shot paraphrasing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 90–121.

Brian Thompson and Matt Post. 2020b. Paraphrase generation as zero-shot multilingual translation: Disentangling semantic similarity from lexical and syntactic diversity. In *Proceedings of the Fifth Conference on Machine Translation*, pages 561–570.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in Neural Information Processing Systems*, 30:5998–6008.

Oriol Vinyals, Meire Fortunato, and Navdeep Jaitly. 2015. Pointer networks. In *Proceedings of the 28th International Conference on Neural Information Processing Systems-Volume 2*, pages 2692–2700.

Gerson Vizcarra and Jose Ochoa-Luna. 2020. Paraphrase generation via adversarial penalizations. In *Proceedings of the Sixth Workshop on Noisy User-generated Text (W-NUT 2020)*, pages 249–259.

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

John Wieting and Kevin Gimpel. 2018. Paranmt-50m: Pushing the limits of paraphrastic sentence embeddings with millions of machine translations. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 451–462.

Sam Wiseman and Alexander M Rush. 2016. Sequence-to-sequence learning as beam-search optimization. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1296–1306.

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

Sander Wubben, Antal Van Den Bosch, and Emiel Krahmer. 2010. Paraphrase generation as monolingual translation: Data and evaluation. In *Proceedings of the 6th International Natural Language Generation Conference*.

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.

Lantao Yu, Weinan Zhang, Jun Wang, and Yong Yu. 2017. Seqgan: Sequence generative adversarial nets with policy gradient. In *Proceedings of the AAAI conference on artificial intelligence*, volume 31.

Jianing Zhou, Hongyu Gong, and Suma Bhat. 2021a. PIE: A parallel idiomatic expression corpus for idiomatic sentence generation and paraphrasing. In *Proceedings of the 17th Workshop on Multiword Expressions (MWE 2021)*, pages 33–48, Online. Association for Computational Linguistics.

Jianing Zhou, Hongyu Gong, Srihari Nanniyur, and Suma Bhat. 2021b. From solving a problem boldly to cutting the gordian knot: Idiomatic text generation. *arXiv preprint arXiv:2104.06541*.

Shuguang Zhu, Xiang Cheng, Sen Su, and Shuang Lang. 2017. Knowledge-based question answering by jointly generating, copying and paraphrasing. In *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*, pages 2439–2442.