Learning to Selectively Learn for Weakly-supervised Paraphrase Generation

Kaize Ding[1]* Dingcheng Li[2] Alexander Hanbo Li[3]
Xing Fan[2] Chenlei Guo[2] Yang Liu[2] Huan Liu[1]
[1]Arizona State University [2]Amazon Alexa AI [3]Amazon AWS AI
{kaize.ding,huan.liu}@asu.edu
{lidingch,hanboli,fanxing,guochenl,yangliud}@amazon.com

Abstract

Paraphrase generation is a longstanding NLP task that has diverse applications for downstream NLP tasks. However, the effectiveness of existing efforts predominantly relies on large amounts of golden labeled data. Though unsupervised endeavors have been proposed to address this issue, they may fail to generate meaningful paraphrases due to the lack of supervision signals. In this work, we go beyond the existing paradigms and propose a novel approach to generate high-quality paraphrases with weak supervision data. Specifically, we tackle the weakly-supervised paraphrase generation problem by: (1) obtaining abundant weakly-labeled parallel sentences via retrieval-based pseudo paraphrase expansion; and (2) developing a meta-learning framework to progressively select valuable samples for fine-tuning a pre-trained language model, i.e., BART, on the sentential paraphrasing task. We demonstrate that our approach achieves significant improvements over existing unsupervised approaches, and is even comparable in performance with supervised state-of-the-arts.

1 Introduction

Paraphrase generation is a fundamental NLP task that restates text input in a different surface form while preserving its semantic meaning. It serves as a cornerstone in a wide spectrum of NLP applications, such as question answering (Dong et al., 2017), machine translation (Resnik et al., 2010), and semantic parsing (Berant and Liang, 2014). With the recent advances of neural sequence-to-sequence (Seq2Seq) architecture in the field of language generation, a growing amount of literature has also applied Seq2Seq models to the sentential paraphrasing task.

Despite their promising results, collecting large amounts of parallel paraphrases is often time-consuming and requires intensive domain knowledge. Therefore, the performance of supervised methods could be largely limited in real-world scenarios. Due to this problem, unsupervised paraphrase generation has recently received increasing attention, but the development is still in its infancy. Generally, sampling-based or editing-based approaches (Bowman et al., 2016; Miao et al., 2019) fail to incorporate valuable supervised knowledge, resulting in less coherent and controllable generated paraphrases (Liu et al., 2019). In this work, we propose going beyond the existing learning paradigms and investigate a novel research problem – *weakly-supervised paraphrase generation*, in order to push forward the performance boundary of sentential paraphrasing models with low-cost supervision signals.

As an understudied problem, weakly-supervised paraphrase generation is challenging mainly because of the following reasons: **(i)** although weak supervision has been applied in different low-resource NLP tasks (Dehghani et al., 2017; Aprosio et al., 2019), for paraphrase generation, it is unclear how to automatically acquire abundant weak supervision data that contains coherent, fluent and diverse paraphrases; **(ii)** weakly-labeled paraphrases tend to be noisy and are not equally informative for building the generation model (Ren et al., 2018; Li et al., 2019a; Yoon et al., 2020). Hence, selecting valuable parallel sentences from weakly-labeled data is vital for solving the studied problem; and **(iii)** the state-of-the-art paraphrasing methods are predominantly built upon traditional Seq2Seq models, while the necessity of learning from scratch largely magnifies the learning difficulty when dealing with scarce or noisy training data (Guu et al., 2018). Thus it is imperative to seek a more robust and knowledge-intensive backbone for learning with weakly-labeled paraphrases.

To address the aforementioned challenges, we present a novel approach for learning an effective paraphrasing model from weakly-supervised

[*] Work was done as an intern at Amazon Alexa AI.

Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, pages 5930–5940
November 7–11, 2021. ©2021 Association for Computational Linguistics

parallel data. By virtue of a simple yet effective *pseudo paraphrase expansion* module, for each input sentence, we are able to obtain multiple similar sentences without unbearable labeling cost and treat them as paraphrases. To mitigate the inaccurate supervision signals within the weakly-labeled parallel data and build an effective paraphrasing model, we further select valuable parallel instances by proposing a novel framework named *Learning-To-Selectively-Learn* (LTSL). Remarkably, LTSL leverages meta-learning to progressively exert the power of pre-trained language model, i.e., BERT (Devlin et al., 2018) and BART (Lewis et al., 2019), with weakly-labeled paraphrasing data. From a meta-learning perspective, the BERT-based *gold data selector* is meta-learned to select valuable samples from each batch of weakly paired sentences, in order to fine-tune and maximize the performance of the BART-grounded *paraphrase generator*. Afterwards, the paraphrase generation performance change on a small validation set will be used to perform meta-optimization on the data selection meta-policy. This way the two pre-trained components *gold data selector* and *paraphrase generator* in LTSL are able to reinforce each other by continuously learning on a pool of meta-selection tasks. To summarize, the major contribution of this work is three-fold:

- We investigate an understudied research problem: weakly-supervised paraphrase generation, which sheds light on the research of sentential paraphrasing under a low-resource setting.
- We develop a framework LTSL, which is a new attempt of leveraging meta-learning to enhance pre-trained language model on paraphrase generation with costless weak supervision data.
- We conduct extensive experiments to illustrate the superiority of our approach over both supervised and unsupervised state-of-the-art methods on the task of paraphrase generation.

2 Related Work

Supervised Paraphrase Generation. With the fast development of deep learning techniques, neural Seq2Seq models have achieved superior performance over traditional paraphrase generation methods that rely on exploiting linguistic knowledge (McKeown, 1980, 1983) or utilizing statistical machine translation systems (Dolan et al., 2004; Bannard and Callison-Burch, 2005). Supervised paraphrasing methods are widely studied when the ground-truth parallel sentences are available during the training time. Among supervised efforts, Residual LSTM (Prakash et al., 2016) is one of the earliest works based on neural networks. Later on, Li et al. (2018) propose to make use of deep reinforcement learning and Iyyer et al. (2018); Chen et al. (2019a) leverage syntactic structures to produce better paraphrases. More recently, retrieval-augmented generation methods have also been investigated (Hashimoto et al., 2018; Kazemnejad et al., 2020; Lewis et al., 2020) for paraphrase generation and achieved promising performance.

Unsupervised Paraphrase Generation. Due to the burdensome labeling cost of supervised counterparts, unsupervised paraphrasing methods have drawn increasing research attention in the community. Methods based on variational autoencoders (VAE) are first proposed to generate paraphrases by sampling sentences from the learned latent space (Bowman et al., 2016; Bao et al., 2019; Fu et al., 2019), while the generated sentences are commonly less controllable. To tackle this issue, CGMH (Miao et al., 2019) uses Metropolis-Hastings sampling to add constraints on the decoder at inference time. Furthermore, researchers try to improve the generation performance in terms of semantic similarity, expression diversity, and language fluency by using simulated annealing (Liu et al., 2019), syntactic control (Huang and Chang, 2021), or dynamic blocking (Niu et al., 2020). In addition, pre-trained translation models have been explored to generate paraphrases via back-translation (Wieting et al., 2017; Guo et al., 2021). But still, those methods can hardly achieve comparable results with supervised approaches.

Learning with Weak Supervision. The profound success of machine learning systems largely benefits from abundant labeled data, however, their performance has been shown to degrade noticeably in the presence of inaccurate supervision signals (Hendrycks et al., 2018), especially in an adversary environment (Reed et al., 2014). As one of the central problems in weak supervision, learning with noisy labels has received much research attention. Existing directions mainly focus on: estimating the noise transition matrix (Goldberger and Ben-Reuven, 2017; Patrini et al., 2017), designing robust loss functions or using regularizations (Ghosh et al., 2017; Li et al., 2017; Zhang et al., 2020), correcting noisy labels (Tanaka et al., 2018; Zheng et al., 2021) and selecting or reweight-

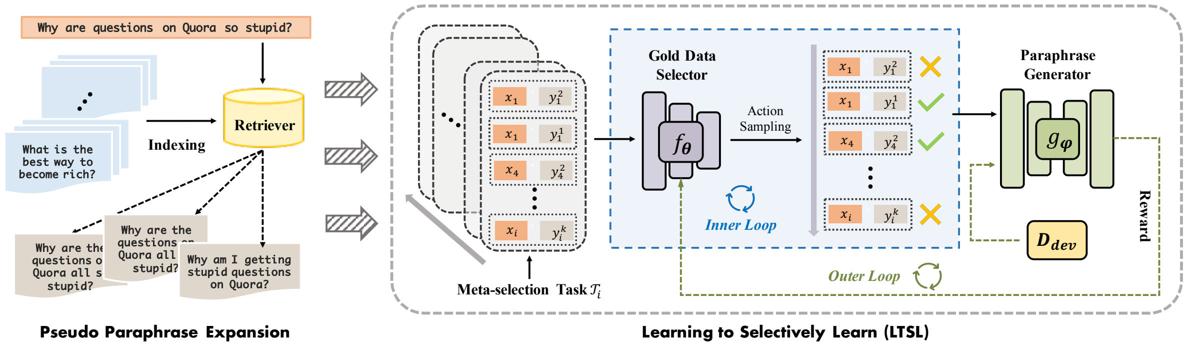

Why are questions on Quora so stupid?
What is the best way to become rich?
Indexing
Retriever
Why are the questions o Quora all s stupid?
Why are the questions on Quora all stupid?
Why am I getting stupid questions on Quora?
Pseudo Paraphrase Expansion
x_1 y_1^2
x_1 y_1^1
x_4 y_4^2
x_i y_i^k
Meta-selection Task $\mathcal{T}_i$
Gold Data Selector
f_θ
Action Sampling
Inner Loop
✗ ✓ ✓ ✗
Paraphrase Generator
g_φ
D_{dev}
Reward
Outer Loop
Learning to Selectively Learn (LTSL)

Figure 1: Overview of our approach for weakly-supervised paraphrase generation. For the LTSL framework, the blue dashed rectangle represents a meta-selection task, while green dashed line is the meta-optimization step.

ing training examples (Ren et al., 2018; Chen et al., 2019b; Yoon et al., 2020). In general, the state-of-the-art methods usually exploit a small clean labelled dataset that is allowed under the low resource setting (Mirzasoleiman et al., 2020). For instance, Gold Loss Correction (Hendrycks et al., 2018) uses a clean validation set to recover the label corruption matrix to re-train the predictor model with corrected labels. Learning to Reweight (Ren et al., 2018) proposes a single gradient descent step guided with validation set performance to reweight the training batch. Learning with noisy/weak supervision has drawn increasing attention in the NLP community (Qin et al., 2018; Feng et al., 2018; Ren et al., 2020), but it is seldomly investigated in the filed of paraphrase generation. In this work, we propose a new meta-learning framework that is capable of selecting valuable instances from abundant retrieved weakly-labeled sentence pairs.

3 Proposed Approach

Figure 1 illustrates our method for solving weakly-supervised paraphrase generation. In essence, there are two sub-tasks: (1) *how to obtain abundant weakly-labeled parallel data from the unlabeled corpus*; and (2) *how to build a powerful paraphrase generation model from noisy weak supervision data*. Formally, given a set of source sentences $\mathcal{X} = \{x_i\}_{i=1}^N$ without ground-truth paraphrases, we first obtain a weakly-labeled parallel corpus $\mathcal{D}_{pseudo} = \{(x_i, y_i)\}_{i=1}^M$ for enabling weak supervision. In this work, we aim to denoise the weak supervision data by selecting a subset of valuable instances $\mathcal{D}_{train} = \{(x_i, y_i)\}_{i=1}^{M'}$ from $\mathcal{D}_{pseudo}$. A small set of trusted parallel sentences $D_{dev} = \{(x_i, y_i)\}_{i=1}^L$ ($L \ll M$) is allowed to be accessed, which is a common assumption in weakly-supervised learning (Ren et al., 2018). In the following subsections, we will introduce how to solve the main challenges with the proposed *pseudo paraphrase expansion* module and the meta-learning framework LTSL.

3.1 Pseudo Paraphrase Expansion

To enable weakly-supervised paraphrase generation, we first propose a plug-and-play *pseudo paraphrase expansion* module. Essentially, the function of this module is to obtain multiple weakly-labeled pseudo paraphrases that are similar or relative to each of the input sequence x.

Expansion via Retrieval. Inspired by the success of retrieval-enhanced methods (Kazemnejad et al., 2020; Lewis et al., 2020) in text generation tasks, we propose to build a retrieval-based expansion module to obtain abundant pseudo parallel paraphrases $\mathcal{D}_{pseudo}$. Given a source sentence x_i, this module automatically retrieves a neighborhood set $\mathcal{N}(x_i)$ consisting of the K most similar sentences $\{y_i^k\}_{k=1}^K$ from a large unlabeled sentence corpus. Specifically, we adopt the simple yet effective retriever BM25 (Robertson and Zaragoza, 2009) in this work. In addition, we use the Elastic Search (Gormley and Tong, 2015) to create a fast search index for efficiently searching for the similar sentences to an input sequence. Here we use the in-domain sentence corpus since it is commonly available in practice and provides better results, but our approach is flexible to be extended to open-domain corpora such as Wikipideia.

Further Discussion. It is worth mentioning that the main reasons of using BM25 rather than a trainable retriever are: (1) this module is not only restricted to retrieval-based expansion, it is designed as a plug-and-play module that can provide more flexibility for weak supervision; and (2) the model training can be more stable since the number of trainable parameters is largely reduced. In addition to the aforementioned retrieval-based method, our approach is also compatible with other expansion alternatives. For instance, we can also adopt domain-adapted paraphraser to generate weakly-labeled paraphrases. Due to the simplicity and learning efficiency, here we focus on retrieval-based expansion to enable weakly-supervised paraphrase generation, and we leave the exploration of other expansion methods for future study.

3.2 Learning to Selectively Learn (LTSL)

The pseudo paraphrase expansion eliminates the dependency of a large amounts of ground-truth labels, nonetheless, one critical challenge is that the obtained weak supervision data is inevitably noisy: though a weakly-labeled paraphrase is somewhat related and convey overlapping information to the input sentence, while they are not parallel in the strict sense. As a result, directly using all the expanded pseudo paraphrase pairs for learning paraphrasing models is unlikely to be effective.

Architecture Overview. To address the aforementioned challenge, we propose a meta-learning framework named Learning-To-Selectively-Learn (LTSL), which is trained to learn data selection meta-policy under weak supervision for building an effective paraphrasing model. LTSL consists of two components: (i) the meta-learner *gold data selector* $f_{\boldsymbol{\theta}}(\cdot)$ parameterized by $\boldsymbol{\theta}$ that determines the selection likelihoods of the training samples to train the base model; and (ii) the base model *paraphrase generator* $g_{\boldsymbol{\phi}}(\cdot)$ with parameters $\boldsymbol{\phi}$, which is a pre-trained autoregressive model that generates a paraphrase given the input sentence. At its core, the meta-learned gold data selector learns to select highly valuable samples from each batch, by measuring their ability to optimize the down-stream paraphrase generator. Meanwhile, the parameters of the paraphrase generator can be updated with the meta-selected samples progressively.

Gold Data Selector (Meta-Learner). In order to represent each pair of parallel sentences, we adopt the widely recognized pre-trained model BERT (Devlin et al., 2018) to build the gold data selector. Specifically, for the i-th weakly supervised paraphrase pair (x_i, y_i), its latent representation can be computed by:

$$\mathbf{z}_i = \text{BERT}([\text{CLS}]\ x_i\ [\text{SEP}]\ y_i\ [\text{SEP}]), \quad (1)$$

where [CLS] and [SEP] are special start and separator tokens. We use the last layer's [CLS] token embedding as $\mathbf{z}_i$. Our gold data selector decides the value of the pair (x_i, y_i) for fine-tuning the pre-trained paraphrase generator by:

$$\mathbf{v}_i = \text{softmax}(\mathbf{W}_s \mathbf{z}_i + \mathbf{b}_s), \quad (2)$$

where both $\mathbf{W}_s$ and $\mathbf{b}_s$ are learnable parameters. Here $\mathbf{v}_i$ is the probability distribution of whether to include the weakly-labeled sentence pair (x_i, y_i).

Paraphrase Generator. The paraphrase generator could be built with any encoder-decoder backbones. In essence, the objective of the paraphrase generator is to maximize the conditional probability $p_{\boldsymbol{\phi}}(y|x)$ over the selected training samples in $\mathcal{D}_{train}$. As pre-trained language models are already equipped with extensive knowledge and have shown strong capability on a diverse set of generation tasks, we propose to use pre-trained language model BART (Lewis et al., 2019) to build our paraphrase generator, which can largely reduce the difficulty of learning from scratch. Specifically, the fine-tuning objective of BART is:

$$J(\boldsymbol{\phi}) = \sum_{(x,y)\in\mathcal{D}_{train}} -\log p_{\boldsymbol{\phi}}(y|x). \quad (3)$$

Meta Reinforcement Learning. Our proposed meta-learning framework LTSL aims to learn a discriminative data selection meta-policy for maximizing the performance of the paraphrase generator. For each batch of data samples in $\mathcal{D}_{pseudo}$, we consider it as a meta-selection task that contains a series of selective actions. As the data selection process is inherently non-differentiable, we adopt reinforcement learning (RL) to enable the meta-optimization. We describe the RL environment of each meta-selection task as follows:

STATE. The state s_t is meant to be a summarization of the learning environment at time step t, which encodes the following information: (i) the representation of the t-th weakly-paired sentences; (ii) the average of the representations of all selected sentences at time step t. The gold data selector will take the concatenated vector as input and output the probability distribution that indicates whether to select this instance or not.

ACTION. At each time step t, the action $a_t \in \{0, 1\}$ decides whether to select the current weakly-labeled instance (x_t, y_t). Ideally, the gold data selector can take the action to select those useful instances for the paraphrasing task and filter out those noisy ones. Specifically, $a_t = 1$ represents the the current weakly-labeled instance will be selected, otherwise (i.e., $a_t = 0$) not. The action on each sentence pair is sampled according to the output of the selection policy $\pi_{\boldsymbol{\theta}}(\cdot|s_t)$.

Reward. The function of reward is to guide the meta-learner to select valuable training instances to improve the performance of the pre-trained generator. After each batch of selections, the accumulated reward of this meta-selection task is determined by the performance change of the paraphrase generator evaluated on the validation set $\mathcal{D}_{dev}$. Note that we use *Perplexity* instead of word-overlapping metrics such as BLEU for evaluation since it is shown to be more efficient and stable (Zhao et al., 2020) for generation tasks. For each meta-selection task, the policy network receives a delayed reward when it finishes all the selections, which is a commonly used design in RL literature (Yao et al., 2019).

3.3 Meta-optimization

To optimize the data selection meta-policy, we aim to maximize the sum of expected rewards of each meta-selection task, which can be formulated as:

$$J(\boldsymbol{\theta}) = \mathbb{E}_{\pi_{\boldsymbol{\theta}}}[\sum_{t=1}^{T} r_t], \tag{4}$$

where r_t is the reward at time step t and $\boldsymbol{\theta}$ is the parameter of the meta-learner gold data selector. We update the $\boldsymbol{\theta}$ via policy gradient:

$$\boldsymbol{\theta}^* \leftarrow \boldsymbol{\theta} + \alpha \nabla_{\boldsymbol{\theta}} \tilde{J}(\boldsymbol{\theta}), \tag{5}$$

where α denotes the learning rate. With the obtained rewards, the gradient can be computed by:

$$\nabla_{\boldsymbol{\theta}} \tilde{J}(\boldsymbol{\theta}) = \sum_{t=1}^{B} r_t \nabla_{\boldsymbol{\theta}} \log \pi_{\boldsymbol{\theta}}(a_t|s_t), \tag{6}$$

where B is the number of instances in a batch. The details of the training process are shown in Algorithm 1. Specifically, we adapt the REINFORCE algorithm (Williams, 1992) to optimize the policy gradients and implement a reward baseline to lower the variance during the training.

In essence, the obtained reward via a small validation set is used to conduct meta-optimization on the data selection meta-policy. By learning on a pool of meta-selection tasks, the meta-learned gold data selector can select valuable instances for enhancing the paraphrase generator. During the meta-learning process, the gold data selector and paraphrase generator can reinforce each other, which progressively improves the data selection policy and enhances the paraphrase generator.

Algorithm 1: Learning algorithm of LTSL

```
Input: Weakly-labeled parallel set $\mathcal{D}_{pseudo}$ and the
       pre-trained language model $g_{\boldsymbol{\phi}}(y|x)$
Output: A paraphrase generation model $g_{\boldsymbol{\phi}}(y|x)$
while $i < Epoch$ do
    // Meta-selection Task $\mathcal{T}_i$
    Sample $B$ samples $\mathcal{D}_B$ from $\mathcal{D}_{pseudo}$
    for $t = 1 \to B$ do
        Compute the state representation $s_t$
        Compute selection probabilities via Eq. (2)
        Sample the action $a_t$ for the current instance
    // Meta-optimization
    Fine-tune the pre-trained generator $g_{\boldsymbol{\phi}}(y|x)$ with
      selected samples to get $g_{\boldsymbol{\phi}'}(y|x)$
    Calculate the reward $r_t$ on the validation set
      between $g_{\boldsymbol{\phi}}(y|x)$ and $g_{\boldsymbol{\phi}'}(y|x)$
    Update $\pi_{\boldsymbol{\theta}}$ according to Eq. (5)
    if $i$ mod $T == 0$ then
        Select data $D_{train}$ from $D_{pseudo}$ using $\pi_{\boldsymbol{\theta}}$
        Update the generator $g_{\boldsymbol{\phi}}(y|x)$ with $D_{train}$
return The fine-tuned paraphrase generator $g_{\boldsymbol{\phi}}(y|x)$;
```

4 Experiments

For the evaluation, we briefly introduce the experimental settings and conduct extensive experiments to corroborate the effectiveness of our approach. The details of our experimental settings and implementations can be found in the Appendix.

4.1 Experimental Settings

Evaluation Datasets & Metrics. In our experiments, we evaluate our proposed approach on multiple widely used paraphrasing benchmark datasets. Since the problem of weakly-supervised paraphrase generation remains largely under-studied in the community, we compare our approach with both supervised and unsupervised paraphrase generation methods. It is worth mentioning that, *due to historical reasons, existing supervised and unsupervised methods use different data splits and evaluation metrics*. To make a fair and comprehensive evaluation, we follow the setting of each line of work

Table 1: Statistics of evaluation datasets

Datasets	Train	Valid	Test	Corpus	Vocab
Quora-S	100K	3K	30K	400K	8K
Twitter	110K	1K	5K	670K	8K
Quora-U	117K	3K	20K	400K	8K
MSCOCO	110K	10K	40K	500K	10K

Table 2: Performance results of all the baseline methods on different paraphrasing datasets.

	Method	Quora-S				Twitter			
		BLEU-2	BLEU-4	ROUGE-1	ROUGE-2	BLEU-2	BLEU-4	ROUGE-1	ROUGE-2
Supervised	Res-LSTM	38.52	24.56	59.69	32.71	32.13	25.92	41.77	27.94
	Transformer	42.91	30.38	61.25	34.23	40.34	32.14	44.53	29.55
	RbM	43.54	-	38.11	32.84	44.67	-	41.87	24.23
	RaE	40.35	25.37	62.71	31.77	44.33	34.16	47.55	31.53
	FSTE	**51.03**	33.46	**66.17**	39.55	46.35	34.62	49.53	32.04
Weakly-supervised	WS-BART	44.19	31.18	58.69	33.39	45.03	34.00	51.34	35.89
	LTSL (ours)	49.18	**36.05**	64.36	**39.71**	**49.30**	**37.94**	**56.02**	**40.61**

	Method	Quora-U				MSCOCO			
		iBLEU	BLEU	ROUGE-1	ROUGE-2	iBLEU	BLEU	ROUGE-1	ROUGE-2
Unsupervised	CGMH	9.94	15.73	48.73	26.12	7.84	11.45	32.19	8.67
	UPSA	12.02	18.18	56.51	30.69	9.26	14.16	37.18	11.21
	PUP	14.91	19.68	59.77	30.47	10.72	15.81	37.38	13.87
	BackTrans	15.51	26.91	52.56	27.85	7.53	10.80	36.12	11.03
	set2seq+RTT	14.66	22.53	59.98	34.09	11.39	17.93	40.28	14.04
Weakly-supervised	WS-BART	17.04	27.63	56.43	33.39	10.91	15.90	40.65	15.62
	LTSL (ours)	**19.20**	**29.25**	**61.71**	**39.21**	**13.45**	**18.87**	**45.18**	**19.17**

and conduct the comparison respectively. Specifically, we use the following datasets to compare with supervised methods:

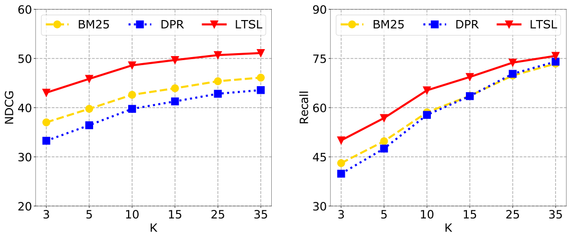
- **Quora-S**: is the Quora question pair dataset which contains 260K non-parallel sentence pairs and 140K parallel paraphrases. Here we denote the version used by supervised methods as Quora-S. We follow the same setting in Li et al. (2018); Kazemnejad et al. (2020) and randomly sample 100K, 30K, 3K parallel sentences for training, test, and validation, respectively.
- **Twitter**: is the twitter URL paraphrasing corpus built by Lan et al. (2017). Following the setting in Li et al. (2018); Kazemnejad et al. (2020), we sample 110K instances from about 670K automatically labeled data as our training set and two non-overlapping subsets of 5K and 1K instances from the human-annotated data for the test and validation sets, respectively.

To compare our approach with unsupervised efforts, we adopt another two benchmark datasets:

- **Quora-U**: is the version of Quora dataset used by unsupervised paraphrasing methods. We follow the setting in Miao et al. (2019); Liu et al. (2019) for a fair comparison and use 3K and 20K pairs for validation and test, respectively.
- **MSCOCO**: is an image captioning dataset containing 500K+ paraphrases pairs for over 120K image captions. We follow the standard splitting (Lin et al., 2014) and evaluation protocols (Liu et al., 2019) in our experiments.

The detailed dataset statistics are summarized in Table 1. Notably, although all the datasets have ground-truth paraphrases, our approach does not use them in the training set, which is as same as unsupervised methods (Siddique et al., 2020). We only allow the model to access the parallel sentences in the validation set during the learning process. Specifically, when comparing with supervised baselines, we follow the previous works and adopt BLEU-n (Papineni et al., 2002) (up to n-grams), and ROUGE (Lin, 2004) scores as evaluation metrics; similarly, we use iBLEU (Sun and Zhou, 2012), BLEU (Post, 2018) and ROUGE scores for comparing with unsupervised methods.

Compared Methods. To show the superiority of our approach, we first include both widely used and state-of-the-art paraphrase generation methods as our baselines. In general, those methods can be divided into two categories: (1) *supervised* methods that are trained with all the parallel sentences in the training corpus, including **Residual LSTM** (Prakash et al., 2016), **Transformer** (Vaswani et al., 2017), **RbM** (Li et al., 2018), and two retrieval-based methods **RaE** (Hashimoto et al., 2018) and **FSTE** (Kazemnejad et al., 2020); (2) *unsupervised* baselines without accessing ground-truth parallel data, including **CGMH** (Miao et al., 2019), **UPSA** (Liu et al., 2019), **PUP** (Siddique et al., 2020), **BackTrans** (back-translation with pretrained NMT model) and **set2seq+RTT** (Guo et al., 2021).



(a) **NDCG@K** (b) **Recall@K**

Figure 2: Selection effectiveness comparison results.

In addition, we also include another *weakly-supervised* method **WS-BART** where we use the same BART model (Lewis et al., 2019) as in LTSL and directly fine-tune it with the clean validation set. Since this model is only fine-tuned with limited labeled data, here we consider it as a *weakly-supervised* baseline.

4.2 Automatic Evaluation Results

Paraphrase Generation. Table 2 summarizes the paraphrasing results of different methods. Overall, the results show that our approach LTSL achieves the state-of-the-art performance in most metrics. Specifically, we can make the following observations from the table:

- LTSL outperforms most of the *supervised* baselines and achieves comparable performance to the state-of-the-art method (i.e., FSTE). In contrast with those supervised baselines that require large amounts of labeled parallel paraphrases, our approach LTSL delivers promising results with very low-cost supervision signals. This enables us to build an effective paraphrasing model under the real-world low-resource setting.
- Compared to *unsupervised* approaches, LTSL overall achieves large performance improvements, especially on iBLEU and BLEU scores. The main reason is that the sampling or editing mechanisms in those methods lack supervised knowledge from parallel sentences. As shown in (Niu et al., 2020), those methods even fall behind a simple baseline *copy-input* which directly copies the source sentence as the output. Our approach is able to well alleviate this weakness by leveraging the knowledge from weak supervision data and pre-trained language models.
- By virtue of the strength of pre-trained language model, the *weakly-supervised* baseline WS-BART performs relatively well compared to existing methods. However, it still falls behind our approach LTSL by a large margin, which demonstrates the effectiveness of our proposed learning framework.

Table 3: Ablation results on Quora-S dataset.

Method	Quora-S			
	BLEU-2	BLEU-4	ROUGE-1	ROUGE-2
pre-trained BART	44.78	31.44	55.35	33.40
w/o selection	44.15	31.98	55.64	33.65
w/o BART	45.79	32.18	57.69	35.39
LTSL	**49.18**	**36.05**	**64.36**	**39.71**

Gold Data Selection. To further evaluate the effectiveness of the meta-learned data selection policy, we use the well-trained gold data selector to compute data values on unseen weakly-supervised candidates during training. Specifically, for each sentence in the test set, we retrieve 50 most similar sentences and consider them as weakly-labeled paraphrases, and then the gold data selector is used to predict the quality of these paraphrase candidates. Note that here we also include the adopted retriever, BM25, and a BERT-based Dense Passage Retriever (DPR) fine-tuned with the labeled parallel sentences from validation set for comparison. We use the computed probabilities to rank all the candidates and report NDCG@K and Recall@K in Figure 2 (a) and (b), respectively. We can see from the figures that the meta-learned gold data selector from LTSL is able to better select ground-truth parallel data than BM-25 and DPR. This observation indicates that the meta-learned gold data selector can effectively generalize to unseen data and select valuable weakly-labeled instances.

Ablation & Parameter Study. Next, we conduct ablation experiments to investigate the contributions of each component in the proposed approach. From the results reported in Table 3, we can learn that by removing the gold data selector, the variant *w/o selection* only achieves similar results with *pre-trained BART* (a variant without pseudo paraphrase

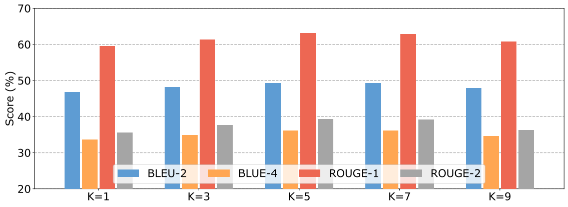


Figure 3: Parameter analysis (K) on Quora-S dataset.

Table 4: Examples of the generated paraphrases on Quora-U dataset. We highlight the key phrases in the paraphrases generated by each method, and we use underline to show the matched parts between LTSL and reference.

Input	BackTrans	WS-BART	LTSL	Reference
How do I attract contributors for my project on Github?	How do I attract contributors for my Github project?	How do I get my project up and running on Github?	How do I get more contributors on my GitHub?	How do I can get more contributors on GitHub?
Is Azusa Pacific University accepting LGBT people?	Does Azusa Pacific University accept LGBT people?	How accepting is Azusa Pacific University of LGBT people?	Should I worry about attending Azusa Pacific University if I am LGBT?	Should I worry about attending Azusa Pacific University if I am gay?
What are the strangest facts about some famous Bollywood movies?	What are the strangest facts about some famous Bollywood movies?	What are the strangest or weirdest facts about Bollywood movies?	What are some of the strangest facts about famous Bollywood movies?	What are some of the strangest facts about famous Bollywood movies?
What could be the reason behind Arnab Goswami quitting Times Now?	What can be the reason of Arnab Goswami quitting Times Now?	What is the reason of Arnab Goswami leaving Times Now?	Why did Arnab Goswami resign from Times Now?	Why Arnab Goswami resigned Times Now?

expansion and gold data selection). It again verifies the importance of learning effective data selection meta-policy. Meanwhile, by using a vanilla Transformer to build the paraphrase generator (w.r.t., *w /o BART*), the performance falls behind LTSL by a considerable gap, which shows the necessity of leveraging pre-trained language models.

We also examine the effect of parameter K on the final performance and show the results on the Quora-S dataset in Figure 3 (similar results can be observed for other datasets). As we can see from the results, with the growth of K, the performance reaches the peak when K is set to 5 and then gradually decreases if K increases. It shows that it is necessary to incorporate abundant weakly-labeled data. However, when more candidates are added, the introduced noise from weakly-labeled data could impair the final performance.

Table 5: Human evaluation results on Quora-U dataset.

Method	Coherence		Fluency		Diversity	
	Score	κ	Score	κ	Score	κ
BackTrans	3.86	0.44	3.95	0.48	3.09	0.38
WS-BART	4.04	0.49	4.60	0.51	3.25	0.43
LTSL	**4.39**	**0.57**	**4.84**	**0.63**	**3.54**	**0.48**

4.3 Human Evaluation Results

To further illustrate the superior quality of the paraphrases generated by LTSL, we conduct subjective human evaluations. We randomly select 100 sentences from the Quora-U dataset and ask three human annotators to evaluate the top three performing methods under the unsupervised paraphrase generation setting. Table 5 presents the average scores along with the inter-annotator agreement (measured by Cohen's kappa κ) in terms of semantic coherence, language fluency, and expression diversity. We rate the paraphrases on a scale of 1-5 (1 being the worst and 5 the best) for the three evaluation criteria. As shown in the table, our approach outperforms all the competing approaches in terms of all the three perspectives. Moreover, the inter-annotator agreement shows moderate or good agreement between raters when assessing the outputs of our model.

4.4 Case Study

Last, we showcase the generated paraphrases from different methods on the Quora-U dataset. As illustrated in Table 4, we can clearly see qualitatively that LTSL can produce more reasonable paraphrases than the other methods in terms of both closeness in meaning and difference in expressions. For example, "How do I attract contributors for my project on Github?" is paraphrased as "How do I get more contributors on my GitHub project?". It is worth mentioning that existing unsupervised methods such as BackTrans cannot generate high-quality paraphrases in terms of diversity, mainly because of the shortage of supervision signals. On the contrary, our LTSL approach is able to generate highly fluent and diverse paraphrases by leveraging valuable weak supervision data and the knowledge of large-scale pre-trained language model.

5 Conclusion

In this work we investigate the problem of paraphrase generation under the low-resource setting and propose a weakly-supervised approach. From automatic and human evaluations, we demonstrate that our approach achieves the state-of-the-art results on benchmark datasets. An interesting direction is to improve the generation performance by leveraging weakly-labeled data from different sources. We leave this as future work.

References

Alessio Palmero Aprosio, Sara Tonelli, Marco Turchi, Matteo Negri, and Mattia A Di Gangi. 2019. Neural text simplification in low-resource conditions using weak supervision. In *ACL NeuralGen Workshop*.

Colin Bannard and Chris Callison-Burch. 2005. Paraphrasing with bilingual parallel corpora. In *ACL*.

Yu Bao, Hao Zhou, Shujian Huang, Lei Li, Lili Mou, Olga Vechtomova, Xinyu Dai, and Jiajun Chen. 2019. Generating sentences from disentangled syntactic and semantic spaces. In *ACL*.

Jonathan Berant and Percy Liang. 2014. Semantic parsing via paraphrasing. In *ACL*.

Samuel Bowman, Luke Vilnis, Oriol Vinyals, Andrew Dai, Rafal Jozefowicz, and Samy Bengio. 2016. Generating sentences from a continuous space. In *SIGNLL*.

Mingda Chen, Qingming Tang, Sam Wiseman, and Kevin Gimpel. 2019a. Controllable paraphrase generation with a syntactic exemplar. In *ACL*.

Pengfei Chen, Ben Ben Liao, Guangyong Chen, and Shengyu Zhang. 2019b. Understanding and utilizing deep neural networks trained with noisy labels. In *ICML*.

Wei Chen, Tie-Yan Liu, Yanyan Lan, Zhi-Ming Ma, and Hang Li. 2009. Ranking measures and loss functions in learning to rank. In *NeurIPS*.

Mostafa Dehghani, Hamed Zamani, Aliaksei Severyn, Jaap Kamps, and W Bruce Croft. 2017. Neural ranking models with weak supervision. In *SIGIR*.

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

Bill Dolan, Chris Quirk, and Chris Brockett. 2004. Unsupervised construction of large paraphrase corpora: Exploiting massively parallel news sources. In *COLING*.

Li Dong, Jonathan Mallinson, Siva Reddy, and Mirella Lapata. 2017. Learning to paraphrase for question answering. In *EMNLP*.

Jun Feng, Minlie Huang, Li Zhao, Yang Yang, and Xiaoyan Zhu. 2018. Reinforcement learning for relation classification from noisy data. In *AAAI*.

Yao Fu, Yansong Feng, and John P Cunningham. 2019. Paraphrase generation with latent bag of words. In *NeurIPS*.

Aritra Ghosh, Himanshu Kumar, and PS Sastry. 2017. Robust loss functions under label noise for deep neural networks. In *AAAI*.

Jacob Goldberger and Ehud Ben-Reuven. 2017. Training deep neural-networks using a noise adaptation layer. In *ICLR*.

Clinton Gormley and Zachary Tong. 2015. Elasticsearch: The definitive guide.

Zilu Guo, Zhongqiang Huang, Kenny Q Zhu, Guandan Chen, Kaibo Zhang, Boxing Chen, and Fei Huang. 2021. Automatically paraphrasing via sentence reconstruction and round-trip translation. In *IJCAI*.

Kelvin Guu, Tatsunori B Hashimoto, Yonatan Oren, and Percy Liang. 2018. Generating sentences by editing prototypes. In *TACL*.

Tatsunori B Hashimoto, Kelvin Guu, Yonatan Oren, and Percy S Liang. 2018. A retrieve-and-edit framework for predicting structured outputs. In *NeurIPS*.

Dan Hendrycks, Mantas Mazeika, Duncan Wilson, and Kevin Gimpel. 2018. Using trusted data to train deep networks on labels corrupted by severe noise. In *NeurIPS*.

Kuan-Hao Huang and Kai-Wei Chang. 2021. Generating syntactically controlled paraphrases without using annotated parallel pairs. In *EACL*.

Mohit Iyyer, John Wieting, Kevin Gimpel, and Luke Zettlemoyer. 2018. Adversarial example generation with syntactically controlled paraphrase networks. In *NAACL*.

Amirhossein Kazemnejad, Mohammadreza Salehi, and Mahdieh Soleymani Baghshah. 2020. Paraphrase generation by learning how to edit from samples. In *ACL*.

Wuwei Lan, Siyu Qiu, Hua He, and Wei Xu. 2017. A continuously growing dataset of sentential paraphrases. In *EMNLP*.

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*.

Patrick Lewis, Ethan Perez, Aleksandara Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, et al. 2020. Retrieval-augmented generation for knowledge-intensive nlp tasks. *arXiv preprint arXiv:2005.11401*.

Junnan Li, Yongkang Wong, Qi Zhao, and Mohan S Kankanhalli. 2019a. Learning to learn from noisy labeled data. In *CVPR*.

Yuncheng Li, Jianchao Yang, Yale Song, Liangliang Cao, Jiebo Luo, and Li-Jia Li. 2017. Learning from noisy labels with distillation. In *ICCV*.

Zichao Li, Xin Jiang, Lifeng Shang, and Hang Li. 2018. Paraphrase generation with deep reinforcement learning. In *EMNLP*.

Zichao Li, Xin Jiang, Lifeng Shang, and Qun Liu. 2019b. Decomposable neural paraphrase generation. In *ACL*.

Chin-Yew Lin. 2004. Rouge: A package for automatic evaluation of summaries. In *ACL Workshop*.

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 *ECCV*.

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

Kathleen McKeown. 1983. Paraphrasing questions using given and new information. *American Journal of Computational Linguistics*.

Kathleen R McKeown. 1980. Paraphrasing using given and new information in a question-answer system. *Technical Reports (CIS)*.

Ning Miao, Hao Zhou, Lili Mou, Rui Yan, and Lei Li. 2019. Cgmh: Constrained sentence generation by metropolis-hastings sampling. In *AAAI*.

Baharan Mirzasoleiman, Kaidi Cao, and Jure Leskovec. 2020. Coresets for robust training of neural networks against noisy labels. In *NeurIPS*.

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 *ACL*.

Giorgio Patrini, Alessandro Rozza, Aditya Krishna Menon, Richard Nock, and Lizhen Qu. 2017. Making deep neural networks robust to label noise: A loss correction approach. In *CVPR*.

Matt Post. 2018. A call for clarity in reporting bleu scores. *arXiv preprint arXiv:1804.08771*.

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. In *COLING*.

Pengda Qin, Weiran Xu, and William Yang Wang. 2018. Robust distant supervision relation extraction via deep reinforcement learning. In *ACL*.

Scott Reed, Honglak Lee, Dragomir Anguelov, Christian Szegedy, Dumitru Erhan, and Andrew Rabinovich. 2014. Training deep neural networks on noisy labels with bootstrapping. *arXiv preprint arXiv:1412.6596*.

Mengye Ren, Wenyuan Zeng, Bin Yang, and Raquel Urtasun. 2018. Learning to reweight examples for robust deep learning. In *ICML*.

Wendi Ren, Yinghao Li, Hanting Su, David Kartchner, Cassie Mitchell, and Chao Zhang. 2020. Denoising multi-source weak supervision for neural text classification. In *EMNLP*.

Philip Resnik, Olivia Buzek, Chang Hu, Yakov Kronrod, Alex Quinn, and Benjamin B Bederson. 2010. Improving translation via targeted paraphrasing. In *EMNLP*.

Stephen Robertson and Hugo Zaragoza. 2009. *The probabilistic relevance framework: BM25 and beyond*. Now Publishers Inc.

AB Siddique, Samet Oymak, and Vagelis Hristidis. 2020. Unsupervised paraphrasing via deep reinforcement learning. In *KDD*.

Hong Sun and Ming Zhou. 2012. Joint learning of a dual smt system for paraphrase generation. In *ACL*.

Daiki Tanaka, Daiki Ikami, Toshihiko Yamasaki, and Kiyoharu Aizawa. 2018. Joint optimization framework for learning with noisy labels. In *CVPR*.

Jörg Tiedemann and Santhosh Thottingal. 2020. Opus-mt–building open translation services for the world. In *EAMT*.

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 *NeurIPS*.

John Wieting, Jonathan Mallinson, and Kevin Gimpel. 2017. Learning paraphrastic sentence embeddings from back-translated bitext. In *EMNLP*.

Ronald J Williams. 1992. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine learning*.

Ziyu Yao, Jayavardhan Reddy Peddamail, and Huan Sun. 2019. Coacor: Code annotation for code retrieval with reinforcement learning. In *The Web Conference*.

Jinsung Yoon, Sercan Arik, and Tomas Pfister. 2020. Data valuation using reinforcement learning. In *ICML*.

Zizhao Zhang, Han Zhang, Sercan O Arik, Honglak Lee, and Tomas Pfister. 2020. Distilling effective supervision from severe label noise. In *CVPR*.

Mingjun Zhao, Haijiang Wu, Di Niu, and Xiaoli Wang. 2020. Reinforced curriculum learning on pretrained neural machine translation models. *arXiv preprint arXiv:2004.05757*.

Guoqing Zheng, Ahmed Hassan Awadallah, and Susan Dumais. 2021. Meta label correction for noisy label learning. In *AAAI*.

A Appendix

A.1 Evaluation Details

For all the evaluation datasets, we follow previous research to pre-process the datasets. For the comparison with supervised methods, we adopt two benchmark datasets (i.e., Quora-S[1] and Twitter[2]) and truncate sentences in both of the datasets to 20 tokens as in Li et al. (2018, 2019b). For the comparison with unsupervised methods, we use Quora-U and MSCOCO[3]. Due to the space limit and data quality issues (Niu et al., 2020), other datasets such as WikiANswers, Twitter (unsupervised version) are not included in our evaluations. We follow their settings and all the sentences are lower cased, and truncate all sentences to up to 20 words.

Throughout the paper, we use those evaluation metrics that have been widely used in the previous work to measure the quality of the paraphrases. In general, BLEU measures how much the words (and/or n-grams) in the machine generated summaries appeared in the human reference summaries. Rouge measures how much the words (and/or n-grams) in the human reference summaries appeared in the machine generated summaries. Specifically, we use the library[4] from HuggingFace to compute BLEU scores and *py-rouge*[5] to compute ROUGE scores. As BLEU and ROUGE could not measure the diversity between the generated and the original sentences, we follow unsupervised paraphrasing methods and adopt iBLEU (Sun and Zhou, 2012) to measure the diversity of expression in the generated paraphrases by penalizing copying words from input sentences. Specifically, we follow the unsupervised paraphrase generation baselines and set the balancing parameter $\alpha = 0.9$.

A.2 Implementation Details

Implementation of LTSL. The proposed model LTSL is trained across 8, 32GB NVIDIA V100 GPUs via distributed training and its inference can be run on one GPU. The batch size B is set to be 512 for all the datasets. The parameter K for pseudo paraphrase expansion is set to 5, T for paraphrase generator update is set to 10 for all the datasets. We use the BERT-base (Devlin et al., 2018) to build our gold data selector, with 12-layer transformer blocks, 768-dimension hidden state, 12 attention heads and total 110M parameters. Specifically, we use the pre-trained BERT-Base-Uncased version. For the paraphrase generator, we use BART-large (Lewis et al., 2019), a pre-trained seq2seq transformer with 400M parameter. For training stage, we use Adam optimizer for fine-tuning with β_1 as 0.9, β_2 as 0.999. The max sequence length of BERT input is set to 66. For learning LTSL, we train the model for 10000 epochs with early-stopping strategy. We grid search for the learning rate in $\{0.0001, 0.0005, 0.001, 0.005, 0.01, 0.05, 0.1\}$, L2 regularization in $\{10^{-6}, 10^{-5}, 10^{-4}, 10^{-3}, 10^{-2}, 10^{-1}\}$ and the dropout rate in $\{0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7\}$. The optimal values are selected when the model achieves the highest accuracy for the validation samples.

Model Pre-training. Pre-training is widely used by reinforcement learning based methods to accelerate the training of RL agent (Yoon et al., 2020; Zhao et al., 2020). Specifically, we use pair-wise ranking loss (Chen et al., 2009) to pre-train the gold data selector using the retrieved data from BM25. This way we can make sure the labeled paraphrases will not leak duing the model pre-training stage.

Implementation of Baselines. Since many recent works on paraphrase generation have not released their implementations, we follow the same data preprocess procedure and the train/test split as used in their papers to make a fair comparison. This way for the supervised and unsupervised baseline methods, we can directly adopt the reported in their papers. Specifically, for the comparison on two datasets Quora-S and Twitter, the results of Residual LSTM, Transformer, RbM, RaE and FSTE are from (Li et al., 2018; Kazemnejad et al., 2020); For the comparison on Quora-U and MSCOCO, we adopt the results of CGMH, UPSA, PUP and set2seq+RTT from (Liu et al., 2019; Siddique et al., 2020; Guo et al., 2021). For BackTrans, we use the Opus-MT (Tiedemann and Thottingal, 2020) to conduct EN-FR and FR-EN back-translation. For the results of human evaluation (Table 5) and case study (Table 4), we run the implementation of UPSA[6] published with the paper and adopt the same parameters as described in the paper (Liu et al., 2019).

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

[2] https://languagenet.github.io/

[3] https://cocodataset.org/#home

[4] https://huggingface.co/metrics/sacrebleu

[5] https://pypi.org/project/py-rouge/

[6] https://github.com/Liuxg16/UPSA