Unsupervised Paraphrasing with Pretrained Language Models

Tong Niu **Semih Yavuz** **Yingbo Zhou**

Nitish Shirish Keskar **Huan Wang** **Caiming Xiong**

Salesforce Research

{tniu, syavuz, yingbo.zhou,
nkeskar, huan.wang, cxiong}@salesforce.com

Abstract

Paraphrase generation has benefited extensively from recent progress in the designing of training objectives and model architectures. However, previous explorations have largely focused on supervised methods, which require a large amount of labeled data that is costly to collect. To address this drawback, we adopt a transfer learning approach and propose a training pipeline that enables pre-trained language models to generate high-quality paraphrases in an unsupervised setting. Our recipe consists of task-adaptation, self-supervision, and a novel decoding algorithm named Dynamic Blocking (DB). To enforce a surface form dissimilar from the input, whenever the language model emits a token contained in the source sequence, DB prevents the model from outputting the subsequent source token for the next generation step. We show with automatic and human evaluations that our approach achieves state-of-the-art performance on both the Quora Question Pair (QQP) and the ParaNMT datasets and is robust to domain shift between the two datasets of distinct distributions. We also demonstrate that our model transfers to paraphrasing in other languages without any additional finetuning.

1 Introduction

Paraphrase generation restates text input in a different surface form while preserving its semantics. It has various applications on downstream NLP tasks including text summarization (Cao et al., 2016), semantic parsing (Berant and Liang, 2014), as well as diversifying text generation for user-facing systems such as chatbots. To evaluate model robustness, a paraphraser can be used to generate adversarial examples, which also serve as augmented data to train the target neural networks (Iyyer et al., 2018a). Besides, paraphrasing queries makes Question Answering systems more likely to match with keywords in a knowledge base (Fader et al., 2014; Yin et al., 2015).

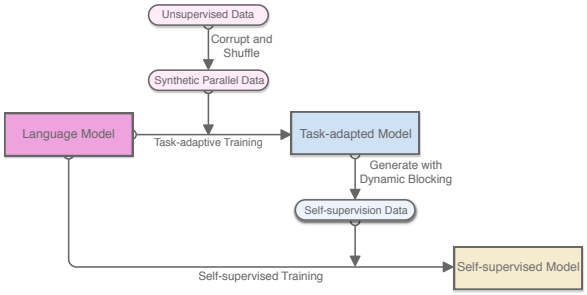

Figure 1: Training pipeline of our paraphrasing model. We first train a task-adapted model with a denoising objective so that it is able to reconstruct input text. We then use Dynamic Blocking (DB) to generate pseudo-pairs of paraphrasing data. Finally, the generated data is used to train the self-supervised model.

However, it is expensive to annotate paraphrases, resulting in only a few human-labeled datasets. The existing ones are either small-scale like MRPC (Dolan and Brockett, 2005), or of closed domains like QQP[1] which consists entirely of questions. Consequently, previous work explored automatically (hence noisily) annotated datasets such as PIT-2015 (Xu et al., 2013), Twitter URL Paraphrase Corpus (Lan et al., 2017), ParaNMT (Wieting and Gimpel, 2018), and ParaBank (Hu et al., 2019), or re-purposed datasets including MSCOCO (Lin et al., 2014) and WikiAnswers (Fader et al., 2013). The scarcity of high-quality datasets motivates us to consider unsupervised alternatives. In this work, we explore a transfer learning approach, which leverages unsupervised large-scale pretrained models like T5 (Raffel et al., 2019) and BART (Lewis et al., 2019).

The effectiveness of BERT-*score* (Zhang et al., 2019) in identifying text similarity hints that pretrained language models are equipped with extensive knowledge in paraphrasing. This knowledge may be attributed to the fact that text spans shar-

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

Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, pages 5136–5150
November 7–11, 2021.

ing similar context usually stay semantically close together – word embedding (Mikolov et al., 2013) being a classic example. In other words, the paraphrasing capability of language models stems from the strong correlation between context and semantic similarity. In this work, we use pre-trained autoregressive LMs to leverage such implicit knowledge for paraphrasing in an unsupervised setting.[2]

For paraphrasing, decoder-only LMs merely output a continuation of the input, while Sequence-to-Sequence models like BART tend to copy the input through even when paired with popular decoding algorithms such as greedy decoding, beam search or top-k/p sampling (Holtzman et al., 2020) because the probabilities of the input tokens during generation are all peaked. To address this issue, we propose *Dynamic Blocking* (DB), a decoding algorithm that effortlessly transforms pre-trained autoregressive language models into natural paraphrasers with the help of *task-adaption* and *self-supervision* (Figure 1). To obtain a surface form different from the input, whenever we emit a token that is present in the source sequence, this algorithm prevents the model from outputting its immediate successor for the next generation step. The algorithm is based on the intuition that during inference, although the top candidate at each generation step corresponds to a peaked probability, the rest of the distribution still contains rich linguistic knowledge suitable for paraphrasing. This is in similar spirit with using soft targets for model distillation (Hinton et al., 2015).

Through automatic and human evaluations, we demonstrate that our approach outperforms previous models (including supervised, in-domain models and the ground-truth targets) on both QQP and ParaNMT datasets and incurs no performance loss under domain shifts (i.e., finetuned on QQP and evaluated on ParaNMT, and vice versa). For automatic evaluations, we propose a *reference-independent* automatic metric named BERT-*i*BLEU, which is a harmonic mean of BERT-*score* and one minus *self*-BLEU. We show that this new metric correlates significantly better with human evaluation than traditional metrics. On the qualitative side, we illustrate with concrete examples that our model generates paraphrases that exhibit diverse syntactic structures. Finally, we observe that our model can generate paraphrases in other languages without any additional training.

[2]We will release all codes.

Our contributions are: (1) a training pipeline that leads to a strong, unsupervised paraphrasing model; (2) a novel decoding algorithm that effectively diversifies paraphrase generation; (3) a new automatic metric that evaluates paraphrasing quality more accurately.

2 Model

Figure 1 shows the training pipeline of our paraphrasing model, which consists of three key components, namely *task-adaptation*, *self-supervision* and *Dynamic Blocking*. Overall we decode the task-adapted model with Dynamic Blocking to generate self-supervision data, which is in turn used to train the final model.

2.1 Task-Adaptation

Inspired by Gururangan et al. (2020), we apply task-adaptive training on the target dataset, treating its training set as a non-parallel collection of sentences. We perform task-adaptation by reconstructing the original sequence from its corrupted version with a denoising auto-encoder objective. Unlike previous work (Devlin et al., 2019; Lewis et al., 2019), we do not corrupt inputs with masks, but rather directly remove the corrupted tokens. This is to avoid pretrain-finetune discrepancy in denoising autoencoding models (Yang et al., 2019). After the deletions, we randomly shuffle all remaining tokens to encourage the model to learn different alignments for better syntactic diversity.[3] Note that we perform both deletions and shuffling on the word-level. This is similar to whole-word masking introduced in later versions of BERT (Devlin et al., 2019). To demonstrate the benefit of our corruption strategy, we present ablation study results in Section 4.3 by either adding masks or not shuffling.

2.2 Dynamic Blocking

Unlike previous diversity-promoting work which mainly focuses on the target side and encourages dissimilarity among beams (Vijayakumar et al., 2018; Kumar et al., 2019; Holtzman et al., 2020), Dynamic Blocking takes the source input into account to guide the model toward generating in a different surface form (Figure 2). As illustrated in Algorithm 1, we represent the source sequence S as a list of tokens $S = (S_0, S_1, ..., S_M)$ and similarly

[3]For example, consider an input sentence "*I want to lose weight in a healthy way.*" where we sample words "*to*" and "*way*" to delete and shuffle the rest. This may give us "*weight in want a lose I healthy .*" as the corrupted sentence.

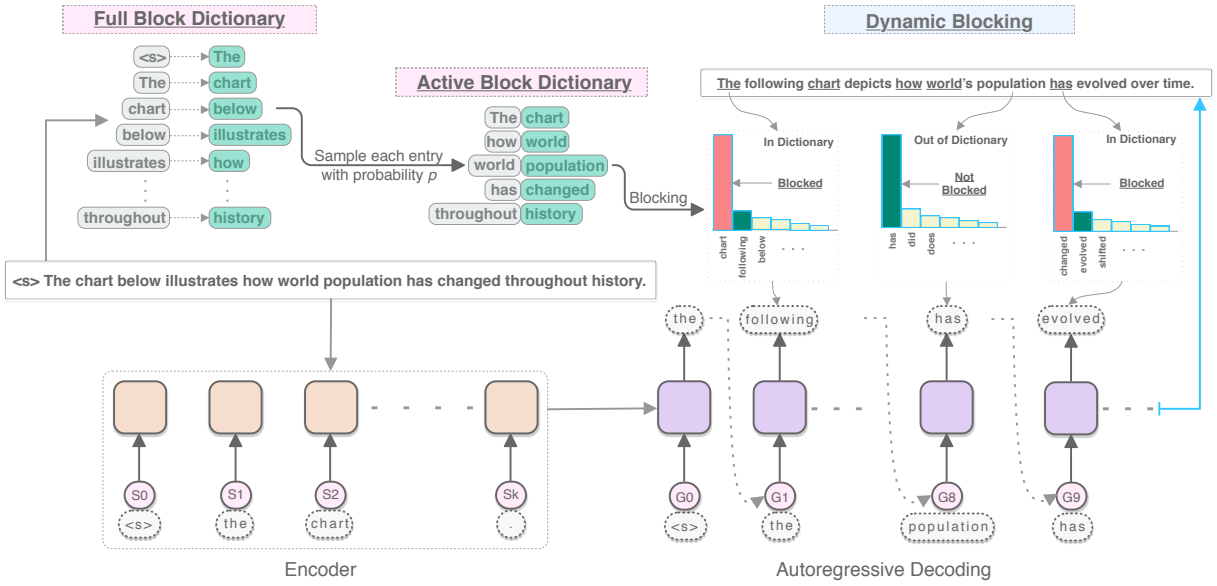

Figure 2: Illustration of the Dynamic Blocking algorithm on real outputs. The algorithm first constructs a *full block dictionary* based on the input, which maps each token to its immediate successor to be blocked, and then samples from this dictionary to build multiple *active block dictionaries*, each used for generating a distinct paraphrase. When establishing an active dictionary, each entry in the full dictionary has a probability of p to be sampled. During generation, the blocking takes place whenever an item in the active dictionary is triggered.

the generated sequence as $G = (G_0, G_1, ..., G_N)$. Suppose that during generation, the model emits G_j that is identical to some S_i (it is not necessary that $i = j$). Then for the next generation step G_{j+1}, the algorithm forbids the model to generate S_{i+1}. Note that we block S_{i+1} for only one step. After G_{j+1} is generated, we perform a different blocking for G_{j+2} iff $G_{j+1} \in S$.

Algorithm 1: Dynamic Blocking

input : A source sequence S consisting of a list of tokens $S = (S_0, S_1, ..., S_M)$, and a $G_0 = \text{BOS}$ to start the decoding process

Initialize $j \leftarrow 0$
while $G_j \neq \text{EOS}$ **do**
  if $G_j = S_i \in S$ *for some* i **then**
    $P(G_{j+1} = S_{i+1} | S, (G_0, G_1, ..., G_j) \leftarrow 0$
  end
  Generate G_{j+1}
  $j \leftarrow j + 1$
end
output : $G = (G_0, G_1, ..., G_N)$

The motivation to block for only one generation step is to allow the possibility of *pure* syntactic variation of the original sequence, meaning that all tokens are kept but their order is permuted. For example, let us consider a decoding algorithm that completely prevents the model from generating a source token at all generation steps – a popular n-gram blocking strategy we call *Static Blocking*. Suppose that we intend to paraphrase "*I like apples and oranges.*" as "*I like oranges and apples.*". This is a valid paraphrase, but if we completely block the word "*apples*" at all generation steps, it will be impossible to arrive at this paraphrase. However, with Dynamic Blocking the model will still be able to generate the word "*apples*" later on even though this word has been temporarily blocked for one step after "*and*" is generated. As shown in Figure 2, Dynamic Blocking builds a block dictionary which maps each token in the source sequence to its immediate successor. We then sample from this dictionary with a probability p for each entry. This hyperparameter controls how different we want the paraphrase to be from the source input. In two extreme cases: when $p = 0.0$, the model does not block any tokens and most likely copies through the source sequence; when $p = 1.0$, the model always blocks the immediate next token, leading to a drastically different surface form. In this work, we take the middle ground and set $p = 0.5$ so that for each blocking action, there will be half of the candidates taking that path. Note that if a word is tokenized into several subwords, *only* the first subword is allowed to be blocked.

We sample multiple block dictionaries to ensure *diversity* among candidates, while leveraging beam search to ensure *coherence*. For each sampled block dictionary, we use beam search to generate four candidates and keep the top-ranked two. It is

beneficial to combine the two decoding methods because beam search helps to weed out ungrammatical or semantically invalid candidates.[4]

Note that we only adopt bi-gram blocking because it is a superset of all higher-gram blockings. Consider, e.g., a tri-gram blocking entry $ab \rightarrow c$ in the block dictionary. If this entry is triggered, then the bi-gram blocking entry $b \rightarrow c$ will also have been triggered. Hence we found it unnecessary to include higher-order n-grams.

2.3 Self-Supervision

To help the model internalize patterns learned from task-adaption, we pseudo-label the training set (Siddhant et al., 2020) by decoding the task-adapted model with Dynamic Blocking. Having obtained the self-supervision data, we discard the task-adapted model and start from the pre-trained language model to avoid catastrophic forgetting (Chronopoulou et al., 2019; Chen et al., 2020). We also include reversed data (i.e., swapping source and target) because during task-adaptation the target is always longer than the input, and including reversed data helps to offset this bias of sequence length.

3 Experimental Setup

3.1 BERT-*i*BLEU

To evaluate paraphrasing quality, we propose a new metric named BERT-*i*BLEU which encourages semantic closeness while penalizing surface-form similarity. For semantic closeness we use the unsupervised metric BERT-*score* (Zhang et al., 2019), which leverages a pretrained language model to compute the cosine similarity between each token in the candidate and that in the reference using contextual embeddings.[5] To ensure that the key information (often conveyed through relatively rare words) is retained in the paraphrase, we apply IDF-reweighing on each token.[6] To measure the surface-form dissimilarity, we use one minus *self*-BLEU, where *self*-BLEU is the BLEU score between the source and the candidate. Hence BERT-*i*BLEU (where *i* stands for *inverse*) is a weighted harmonic mean of the BERT-*score* and one minus *self*-BLEU.

$$\text{BERT-}i\text{BLEU} = \left(\frac{\beta * \text{BERT-}score^{-1} + 1.0 * (1 - \textit{self}\text{-BLEU})^{-1}}{\beta + 1.0}\right)^{-1}$$
$$\textit{self}\text{-BLEU} = \text{BLEU}(source, candidate)$$

As an extreme case, though copying through the input leads to a perfect BERT-*score*, $1 - \textit{self}\text{-BLEU} = 0$; hence BERT-*i*BLEU $= 0$. This is the reason that we do not use the BERT-*score* directly to evaluate paraphrases. β is used to control the relative importance between semantic similarity and surface-form dissimilarity. In our experiments we set $\beta = 4.0$ to scale up BERT-*score* so that it has a similar range with *self*-BLEU. Note that because BERT-*i*BLEU is reference-independent, it serves both as a metric to evaluate paraphrasing quality and as a criterion to re-rank generated candidates during task-adaptation and self-supervision.

3.2 Dataset

We evaluate on the Quora Question Pair (QQP) and the ParaNMT datasets. QQP contains 140K question pairs that are marked as a duplicate to each other and 640K non-parallel questions. The sizes of dev and test sets are 3K and 20K, respectively. The ParaNMT dataset was constructed by back-translating sentences in Czech in the CzEng dataset (Bojar et al., 2016). We directly obtained the test set of SOW-REAP from the authors of Goyal and Durrett (2020). To match the size of their training set, for task-adaptation we sample 350K non-parallel sentences from ParaNMT-5M, while to generate self-supervision data we sample 350K sentences from the same corpus as inputs. We filter out any sentences in SOW-REAP's test set to avoid training on test examples.

3.3 Reproduction of Previous Models

For the experiments on QQP we reproduce the supervised Transformer with the pre-trained T5-base model, which is stronger than the usual setting where the paraphraser trains from scratch. We also reproduce the model from Hegde and Patil (2020), which we refer to as CorruptLM. This model is similar to our task-adaptive phase (Section 2.1), except that they corrupt the inputs by removing all stop words rather than a fixed percentage of arbi-

[4] For more details on Dynamic Blocking, please refer to Appendix D.

[5] In early experiments we tried another unsupervised metric Universal Sentence Encoder (Cer et al., 2018) and supervised metrics including RUSE (Shimanaka et al., 2018), Sentence-BERT (Reimers and Gurevych, 2019), and BLEURT (Sellam et al., 2020). We observed that BERT-*score* worked better at evaluating semantic similarity compared to these metrics.

[6] We use the BookCorpus dataset (Zhu et al., 2015) to compute the IDF weights.

trary words.[7] Instead of GPT-2 as used by their work, we use BART which shows stronger results on downstream tasks. The rest of the settings remain the same.[8] For the experiments on ParaNMT we use the SOW-REAP model released by Goyal and Durrett (2020).[9]

3.4 Automatic Evaluation

To evaluate paraphrasing quality, we follow Li et al. (2019) to report iBLEU (Sun and Zhou, 2012), BLEU (Papineni et al., 2002) and ROUGE (Lin, 2004) on QQP, and report BLEU and ROUGE on ParaNMT. Follwing Goyal and Durrett (2020), for ParaNMT both BLEU and ROUGE are calculated by first selecting the candidate that achieves the *best* sentence-level score with the ground-truth, and then compute the corpus-level score of all these candidates. We use *py-rouge*[10] to compute ROUGE and the *Datasets* library from *HuggingFace*[11] to compute BLEU. We also report BERT-*i*BLEU for the models we reproduced.

3.5 Human Evaluation

We conduct human evaluations on MTurk.[12] For each experiment, we compare our model with the strongest models reported in both supervised and unsupervised settings. On QQP, we compare with supervised Transformer, unsupervised CorruptLM, and the ground-truth. On ParaNMT, we compare with SOW-REAP and the ground-truth. To construct holistic human studies, we opt for both head-to-head binary comparison and Likert-scale scoring. The former provides straightforward results on which model is stronger, while the latter is used to consolidate their relative positions.

We only worked with annotators who had completed more than $10K$ assignments, had an approval rate of $> 98\%$, and resided in the US. We also required that the annotators be native English speakers. When comparing between two model outputs based on the same input, we asked the annotators to identify which paraphrase they prefer in terms of overall quality.[13] For each experiment, we randomly sampled 200 examples from the QQP's or ParaNMT's test set and shuffled the order of each example to anonymize the model identities. Each assignment was scored by two annotators.

4 Results

4.1 Human Evaluation

Table 1 and 2 present human evaluation results on our final model compared with other baselines. On QQP our model outperforms both Transformer and CorruptLM. Recall that CorruptLM also leverages a pre-trained language model. This indicates the effectiveness of our training pipeline when holding the LM factor as a constant. On ParaNMT our model outperforms SOW-REAP in both head-to-head and Likert-based evaluations. Moreover, our model outperforms the ground-truth on both datasets. For ParaNMT, the result indicates that our approach also outperforms a supervised round-trip translation baseline since that is how ParaNMT data was generated in the first place. For QQP, we note two reasons why these scores do not indicate that our model can generate paraphrases with human-level quality. First, QQP is human-labeled, not human-generated. Second, QQP annotates duplicate questions rather than paraphrases. Questions referring to the same topic but are not semantically equivalent may still be marked as duplicates.[14]

We use Cohen's Kappa to evaluate the inter-annotator agreement. For head-to-head evaluations, we obtained $kappa = 0.35$, indicating fair agreement. Note that when calculating kappa, we leave out all cases where either of the two annotators gives a "tie" because this usually signifies that they are unsure about which paraphrase is better.

4.2 Advantage of the Proposed Metric

To facilitate a better understanding of the automatic evaluation results, we investigate how each of the automatic metrics correlates with human evaluation. Table 3 shows that BERT-*i*BLEU agrees sig-

[7] Because the original paper did not provide the source of the stop words, we extract the first 252 words from The Corpus of Contemporary American English (Davies, 2010) to match the number.

[8] To encourage the model to output new words in the reconstructed sentence, CorruptLM starts by randomly replacing 20% of the words in the source sequence with synonyms using Syn-net (Miller, 1998) (also applied during inference).

[9] https://github.com/tagoyal/sow-reap-paraphrasing/

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

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

[12] Screenshots of the interfaces used by our MTurk studies are presented in Appendix F.

[13] We intentionally did not ask them to separately evaluate semantic similarity and surface-form diversity because the latter is easy to check with *self*-BLEU.

[14] For instance, the question pair "*I'm 27, is it too late for me to go to medical school?*" and "*How old is too old to start medical school?*" has a positive label even though they do not share the same meaning.

Dataset	**Ours** v.s.	Win(%)	Tie(%)	Loss(%)	W-L(%)
QQP	Transformer	40.75	28.25	31.00	12.50
	CorruptLM	46.00	26.25	27.75	18.00
	Ground-truth	43.00	16.75	40.25	2.75
ParaNMT	SOW-REAP	40.50	28.50	31.00	9.50
	Ground-truth	49.50	14.50	36.00	13.50

Table 1: Head-to-head human evaluation results. Each experiment is performed over 200 samples with 2 annotators each. "**Ours**" stands for the model trained with self-supervision and decoded with Dynamic Blocking. Note that both Transformer and SOW-REAP are supervised models, and we are also comparing our unsupervised model outputs with the ground-truth. "**W-L**" stands for the difference between *Win* and *Loss*.

Dataset	Model		Avg. Score
QQP	Supervised	Transformer	4.04 ± 1.01
	Unsupervised	CorruptLM	3.74 ± 1.26
		Ours	$\mathbf{4.19 \pm 0.99}$
ParaNMT	Supervised	SOW-REAP	3.78 ± 1.15
	Unsupervised	**Ours**	$\mathbf{3.94 \pm 1.09}$

Table 2: Likert-scale human evaluation results. Both averages and standard deviations are reported.

nificantly better with human perceptions. The reason that BLEU does not correlate well with human evaluation is that there are two conflicting objectives. The first comes from keeping the important information, such as named entities, which should be copied verbatim, while the second comes from using different wordings to express the same semantics – the better the model is at this, the lower the BLEU becomes. For a model good at both, the gain in BLEU for matching key entities and the loss for using different wordings cancel each other out, preventing BLEU from faithfully evaluating the paraphrasing quality. Consequently, BLEU is only useful for checking extreme cases: very low or high BLEU usually signals bad paraphrases, but for the middle-ground cases BLEU alone is less indicative. A similar argument holds for ROUGE. In contrast, BERT-*score* encourages the first objective and is not penalized by the second. However, parroting the input will still fool BERT-*score* alone. Hence we pair it with *self*-BLEU to encourage surface-form diversity.

4.3 Automatic Evaluation

On QQP, our model outperforms both the supervised Transformer and the unsupervised CorruptLM on BERT-*i*BLEU (Table 4).[15] Recall that

[15]We tried combining supervised Transformer with DB, and obtained a BERT-iBLEU of 80.1 on QQP, indicating that DB itself is an effective diversity-promoting decoding strategy.

	BERT-*i*BLEU	iBLEU	BLEU	ROUGE-1/2/L
Agree %	**68.9**	39.4	45.3	21.8/5.4/21.4

Table 3: The percentage of times where the ranking given by each metric agrees with that given by human evaluation in the head-to-head studies. Only cases where two annotators agree are counted.

both Transformer and CorruptLM leverage a strong pretrained language model, indicating that the performance gain stems mainly from our proposed pipeline rather than the language model itself. On ParaNMT, our model outperforms the supervised SOW-REAP (Table 5).[16] As ablation studies on task-adaptation and self-supervision, we can see in Table 4 and 5 that our model (TA+SS+DB) beats the one that is either task-adapted only (TA) or self-supervised but decoded without DB (TA+SS), showing that both self-supervision and Dynamic Blocking are crucial to paraphrasing quality.

On the traditional metrics in Table 4, our models also obtain competitive results with the supervised models. However, as we move down to the last row, we see that Copy-input achieves state-of-the-art results on all metrics except BERT-*i*BLEU, indicating that iBLEU, BLEU, and ROUGE scores are not reliable for evaluating paraphrasing quality.[17] In contrast, our best model on BERT-*i*BLEU (TA+SS+DB) achieves much lower iBLEU and BLEU scores as compared to other models, showing the inconsistency between these traditional metrics and human evaluation. We also note one special aspect of Table 5 to make it easier to interpret. Unlike on QQP, the performance of Copy-input on ParaNMT is the lowest among all models. However, we need to take this comparison with a grain of salt because all the other results are based on 10 candidates where only the ones with the highest sentence-level scores are retained. In contrast, Copy-input only has one candidate. Thus Copy-input and the other results are not directly comparable. Plus, SOW-REAP filters the dataset to only include syntactically diverse targets and then splits it into the train, dev and test sets, which makes Copy-input less effective.

4.4 Robustness to Domain Shift

On the ParaNMT dataset, we notice that CorruptLM, when finetuned on non-parallel QQP,

[16]Please refer to Appendix A for results of our model compared with all previous ones on the traditional metrics.

[17]Mao and Lee (2019) also observe that parroting often achieves competitive results.

	Model	BERT-*i*BLEU	iBLEU	BLEU	ROUGE-1	ROUGE-2	ROUGE-L
Supervised	Transformer	68.7	17.0	22.3	55.8	32.3	57.5
Unsupervised	CorruptLM	61.5	12.1	16.8	49.1	26.2	51.7
	TA	76.2	16.0	21.2	61.9	35.1	61.7
	TA+SS	78.9	15.6	20.7	61.5	32.8	60.7
	TA+SS+DB (NMT)	82.5	10.1	14.6	60.1	28.5	58.6
	TA+SS+DB	**83.1**	9.6	14.1	59.9	28.5	58.8
No Model	Copy-input	0.0	**24.3**	**30.4**	**65.7**	**41.7**	**66.5**

Table 4: Automatic evaluation results on QQP. TA = **T**ask-**A**daptation, SS = **S**elf-**S**upervision and DB = **D**ynamic **B**locking. "NMT" stands for model finetuned on non-parallel ParaNMT and evaluated cross-domain on QQP. Both our final model (TA+SS+DB) and the best result for each metric are boldfaced. Please refer to Section A in the Appendix for a comparison with 12 supervised models and 5 unsupervised models from previous work.

	Model	BERT-*i*BLEU	BLEU	ROUGE-1	ROUGE-2	ROUGE-L
Supervised	SOW-REAP	54.2	**30.9**	**62.3**	**40.2**	**61.7**
Unsupervised	CorruptLM (QQP)	39.7	7.6	31.9	11.6	31.6
	TA	72.0	20.2	59.0	32.3	53.8
	TA+SS	74.0	22.9	58.9	33.3	54.1
	TA+SS+DB (QQP)	76.8	22.0	60.1	33.8	54.9
	TA+SS+DB	**78.0**	22.6	59.8	33.2	54.5
No Model	Copy-input	0.0	18.4	54.4	27.2	49.2

Table 5: Automatic evaluation results on ParaNMT. "QQP" stands for models finetuned on non-parallel QQP and evaluated cross-domain on ParaNMT. Note that BLEU and ROUGE scores are based on top-10 candidates where **only the ones with the highest sentence-level scores** are retained for the final score computation.

achieves much worse results than the other models (CorruptLM (QQP) row in Table 5), indicating that it is less robust to domain shift. In contrast, our model achieves similar results compared to the in-domain one under the same setting (TA+SS+DB (QQP) row). Conversely, we also finetune our model on non-parallel ParaNMT and evaluate on QQP (TA+SS+DB (ParaNMT) row in Table 4). We observe that this model again achieves performance similar to that of the in-domain model. These results show that our model may be able to perform task-adaptation using an arbitrary out-of-domain corpus and still work well on the target domain.

4.5 Ablation Studies on Corruption Strategies

During task-adaptation, our corruption strategies involve both deletions and shuffling. In Table 6 we provide ablation study results where we either replace words with masks instead of deleting them or delete words without shuffling. We can see that our delete-and-shuffle strategy achieves the best BERT-*i*BLEU score among the three settings.

	AddMask	NoShuffle	Delete-Shuffle
BERT-*i*BLEU	80.7	81.7	**83.1**

Table 6: Ablation studies on different corruption strategies for task-adaptation on QQP. AddMask stands for the strategy where corrupted words are replaced with MASK tokens; NoShuffle corresponds to "no shuffling" after sentence corruption.

5 Analysis

5.1 Syntactic Diversity

In Table 7, we qualitatively demonstrate paraphrases generated by our model that exhibit syntactic structure variance. Unlike previous work relying on explicit syntactic scaffolding (Goyal and Durrett, 2020), our model achieves syntactic diversity "for free" from shuffling during task-adaptation.[18]

5.2 Generalization to Other Languages

Dynamic Blocking on BART without Finetuning Though we focus on T5 throughout the paper, we do note a *unique ability* of BART: it can

[18] We present in Appendix B that shuffling also makes the model robust to grammar errors, enabling it to paraphrase and perform text normalization at the same time.

Input	Generated paraphrase
We got to spend the rest of the weekend at the track. yeah.	We got to stay at the track for the rest of the weekend. yeah.
Are predictions of the future based on the present too much?	Are future predictions too much based on the present?
What is the best way to reduce belly and arm fat?	What is the easiest way to reduce arm and belly fat?
You can seduce enemy soldiers, though.	You can, though, seduce enemy troops.
Well, why would your buddy be in the shower with you?!	Okay, why would you be in the shower with your friend?!

Table 7: Selected paraphrases generated by our final model that shows syntactic variance at different extents. Only the top candidate is shown for each input.

directly work with Dynamic Blocking to generate paraphrases (i.e., without domain-adaptation and self-supervision), though of lower quality than the self-supervised model. We demonstrate such examples in Appendix D.

Paraphrasing in Other Languages We observe that although BART is trained almost exclusively on English text, it is able to paraphrase in multiple other languages. We adopt the aforementioned BART setting and present an example in German (Table 13 in Appendix E). To our best knowledge, this is the first unsupervised model that can paraphrase in a non-English language. The reasoning behind this observation is twofold. First, although BART was trained on English corpora, there is a small portion of the content in German due to mislabeled language identification, allowing the model to observe German data; second, previous work has shown that large-scale language models are able to perform zero-shot cross-lingual transfer on a variety of downstream classification tasks, such as Named Entity Recognition (Moon et al., 2019), Natural Language Inference, and Document Classification (Artetxe and Schwenk, 2019). Our work hence demonstrates that it is possible to perform such a transfer even for generative tasks like paraphrasing. We also hypothesize that the paraphrasing quality should improve if we apply our training pipeline to mBART or mT5 (Xue et al., 2020). We leave this as future work.

6 Related Work

Paraphrase generation has been a long-standing task that has several applications on downstream NLP tasks including text summarization (Cao et al., 2016), semantic parsing (Berant and Liang, 2014), and question answering (Yu et al., 2018). Early works on paraphrase generation mostly rely on rule-based or statistical machine translation systems (McKeown, 1980; Meteer and Shaked, 1988; Bannard and Callison-Burch, 2005).

Supervised Approaches Neural sequence-to-sequence (Seq2Seq) models have been used to address this task (Prakash et al., 2016; Li et al., 2017; See et al., 2017; Vaswani et al., 2017; Gupta et al., 2018); sometimes such models are also used to evaluate paraphrasing quality (Thompson and Post, 2020). Round-trip translation between two languages (i.e., back-translation) with strong neural machine translation (NMT) models has also become a widely used approach for paraphrase generation (Yu et al., 2018). Consequently, supervised models using datasets like ParaNMT obtain their performance mainly from sequence-level distillation (Kim and Rush, 2016), where the data comes from the underlying supervised translation models. There have been several previous works (Iyyer et al., 2018b; Chen et al., 2019; Li et al., 2019; Kumar et al., 2019; Goyal and Durrett, 2020) that make use of syntactic structures to produce more diverse paraphrases. More recently, Qian et al. (2019) employ distinct generators to produce diverse paraphrases. Retrieval-augmented generation methods have also been investigated (Kazemnejad et al., 2020; Lewis et al., 2020). However, most of these approaches require parallel data.

Unsupervised Approaches Unsupervised paraphrasing, on the other hand, is a rather less explored and more challenging problem in NLP. Bowman et al. (2016) train a variational autoencoder (VAE) to maximize the lower bounds for the reconstruction log-likelihood of the input sentence without requiring any parallel corpora. Sampling from the trained VAE's decoder leads to sentences that can practically be considered as paraphrases as the decoder aims to reconstruct the input sentence by its training objective. Miao et al. (2018) introduce a constrained sentence generation approach by using Metropolis-Hastings sampling, which allows for decoding with complicated discrete constraints such as the occurrence of multiple keywords, hence not requiring any parallel corpora. Roy and Grangier (2019) introduce a model that allows interpo-

lation from continuous auto-encoders to vector-quantized auto-encoders. Liu et al. (2020) cast the paraphrasing as an optimization problem, where it searches the sentence space to find the optimal point for an objective function that takes semantic similarity, expression diversity, and language fluency into account. Siddique et al. (2020) optimize a similar objective with deep reinforcement learning.

Transfer Learning There have been few works leveraging pre-trained language models for paraphrasing, either in a supervised (Witteveen and Andrews, 2019) or an unsupervised (Hegde and Patil, 2020) setting. Both works employ GPT-2 as their backbone generation model. Similarly, we opt for more recent large-scale pre-trained models like BART and T5.

7 Conclusion

We design an effective training pipeline that enables large-scale pre-trained models to generate high-quality paraphrases in an unsupervised setting through task-adaptation, self-supervision, and a novel decoding algorithm named Dynamic Blocking. We demonstrate with automatic and human evaluations that our model achieves state-of-the-art results on benchmark datasets. We also show that our model generates paraphrases that exhibit syntactic diversity, as well as generalizes to other languages without any additional training. Overall our work motivates a deeper investigation into self-supervised techniques for paraphrase generation as well as extensions such as *context-aware* paraphrasing, where the output conditions not only on the sentences to be paraphrased, but also on the context around them. We leave this as future work.

References

Mikel Artetxe and Holger Schwenk. 2019. Massively multilingual sentence embeddings for zero-shot cross-lingual transfer and beyond. *Transactions of the Association for Computational Linguistics*, 7:597–610.

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

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.

Ondřej Bojar, Ondřej Dušek, Tom Kocmi, Jindřich Libovickỳ, Michal Novák, Martin Popel, Roman Sudarikov, and Dušan Variš. 2016. Czeng 1.6: enlarged czech-english parallel corpus with processing tools dockered. In *International Conference on Text, Speech, and Dialogue*, pages 231–238. Springer.

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

Ziqiang Cao, Chuwei Luo, Wenjie Li, and Sujian Li. 2016. Joint copying and restricted generation for paraphrase. *arXiv preprint arXiv:1611.09235*.

Daniel Cer, Yinfei Yang, Sheng-yi Kong, Nan Hua, Nicole Limtiaco, Rhomni St. John, Noah Constant, Mario Guajardo-Cespedes, Steve Yuan, Chris Tar, Brian Strope, and Ray Kurzweil. 2018. Universal sentence encoder for English. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 169–174, Brussels, Belgium. Association for Computational Linguistics.

Mingda Chen, Qingming Tang, Sam Wiseman, and Kevin Gimpel. 2019. A multi-task approach for disentangling syntax and semantics in sentence representations. 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)*.

Sanyuan Chen, Yutai Hou, Yiming Cui, Wanxiang Che, Ting Liu, and Xiangzhan Yu. 2020. Recall and learn: Fine-tuning deep pretrained language models with less forgetting. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7870–7881, Online. Association for Computational Linguistics.

Alexandra Chronopoulou, Christos Baziotis, and Alexandros Potamianos. 2019. An embarrassingly simple approach for transfer learning from pretrained language models. 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)*.

Mark Davies. 2010. The corpus of contemporary american english as the first reliable monitor corpus of english. *Literary and linguistic computing*, 25(4):447–464.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. 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 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

William B Dolan and Chris Brockett. 2005. Automatically constructing a corpus of sentential paraphrases. In *Proceedings of the Third International Workshop on Paraphrasing (IWP2005)*.

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.

Anthony Fader, Luke Zettlemoyer, and Oren Etzioni. 2014. Open question answering over curated and extracted knowledge bases. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 1156–1165.

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, Online. Association for Computational Linguistics.

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

Suchin Gururangan, Ana Marasović, Swabha Swayamdipta, Kyle Lo, Iz Beltagy, Doug Downey, and Noah A. Smith. 2020. Don't stop pretraining: Adapt language models to domains and tasks. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8342–8360, Online. Association for Computational Linguistics.

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

Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. 2015. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*.

Ari Holtzman, Jan Buys, Li Du, Maxwell Forbes, and Yejin Choi. 2020. The curious case of neural text degeneration. In *The Ninth International Conference on Learning Representations*.

J. Edward Hu, Abhinav Singh, Nils Holzenberger, Matt Post, and Benjamin Van Durme. 2019. Large-scale, diverse, paraphrastic bitexts via sampling and clustering. In *Proceedings of the 23rd Conference on Computational Natural Language Learning (CoNLL)*, pages 44–54, Hong Kong, China. Association for Computational Linguistics.

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

Mohit Iyyer, John Wieting, Kevin Gimpel, and Luke Zettlemoyer. 2018b. 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, New Orleans, Louisiana. Association for Computational Linguistics.

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.

Yoon Kim and Alexander M. Rush. 2016. Sequence-level knowledge distillation. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1317–1327, Austin, Texas. Association for Computational Linguistics.

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, Minneapolis, Minnesota. Association for Computational Linguistics.

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

Mike Lewis, 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*.

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, Online. Association for Computational Linguistics.

Zichao Li, Xin Jiang, Lifeng Shang, and Hang Li. 2017. Paraphrase generation with deep reinforcement learning. *arXiv preprint arXiv:1711.00279*.

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

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

Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. 2014. Microsoft coco: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer.

Xianggen Liu, Lili Mou, Fandong Meng, Hao Zhou, Jie Zhou, and Sen Song. 2020. Unsupervised paraphrasing by simulated annealing. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 302–312, Online. Association for Computational Linguistics.

Hong-Ren Mao and Hung-Yi Lee. 2019. Polly want a cracker: Analyzing performance of parroting on paraphrase generation datasets. 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 5960–5968, Hong Kong, China. Association for Computational Linguistics.

Kathleen R McKeown. 1980. Paraphrasing using given and new information in a question-answer system. *Technical Reports (CIS)*, page 723.

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

Ning Miao, Hao Zhou, Lili Mou, Rui Yan, and Lei Li. 2018. Cgmh: Constrained sentence generation by metropolis-hastings sampling.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.

George A Miller. 1998. *WordNet: An electronic lexical database*. MIT press.

Taesun Moon, Parul Awasthy, Jian Ni, and Radu Florian. 2019. Towards lingua franca named entity recognition with bert. *arXiv preprint arXiv:1912.01389*.

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, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.

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 *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 2923–2934, Osaka, Japan. The COLING 2016 Organizing Committee.

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

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2019. Exploring the limits of transfer learning with a unified text-to-text transformer. *arXiv preprint arXiv:1910.10683*.

Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence embeddings using Siamese BERT-networks. 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 3982–3992, Hong Kong, China. Association for Computational Linguistics.

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, Florence, Italy. Association for Computational Linguistics.

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

Thibault Sellam, Dipanjan Das, and Ankur Parikh. 2020. BLEURT: Learning robust metrics for text generation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7881–7892, Online. Association for Computational Linguistics.

Hiroki Shimanaka, Tomoyuki Kajiwara, and Mamoru Komachi. 2018. RUSE: Regressor using sentence embeddings for automatic machine translation evaluation. In *Proceedings of the Third Conference on Machine Translation: Shared Task Papers*, pages 751–758, Belgium, Brussels. Association for Computational Linguistics.

Aditya Siddhant, Ankur Bapna, Yuan Cao, Orhan Firat, Mia Chen, Sneha Kudugunta, Naveen Arivazhagan, and Yonghui Wu. 2020. Leveraging monolingual data with self-supervision for multilingual neural machine translation. In *Proceedings of the*

58th Annual Meeting of the Association for Computational Linguistics, pages 2827–2835, Online. Association for Computational Linguistics.

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.

Hong Sun and Ming Zhou. 2012. Joint learning of a dual SMT system for paraphrase generation. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 38–42, Jeju Island, Korea. Association for Computational Linguistics.

Brian Thompson and Matt Post. 2020. 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, Online. Association for Computational Linguistics.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need.

Ashwin Vijayakumar, Michael Cogswell, Ramprasaath Selvaraju, Qing Sun, Stefan Lee, David Crandall, and Dhruv Batra. 2018. Diverse beam search for improved description of complex scenes. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32.

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, Melbourne, Australia. Association for Computational Linguistics.

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

Wei Xu, Alan Ritter, and Ralph Grishman. 2013. Gathering and generating paraphrases from twitter with application to normalization. In *Proceedings of the sixth workshop on building and using comparable corpora*, pages 121–128.

Linting Xue, Noah Constant, Adam Roberts, Mihir Kale, Rami Al-Rfou, Aditya Siddhant, Aditya Barua, and Colin Raffel. 2020. mt5: A massively multilingual pre-trained text-to-text transformer. *arXiv preprint arXiv:2010.11934*.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Russ R Salakhutdinov, and Quoc V Le. 2019. Xlnet: Generalized autoregressive pretraining for language understanding. In *Advances in neural information processing systems*, pages 5753–5763.

Pengcheng Yin, Nan Duan, Ben Kao, Junwei Bao, and Ming Zhou. 2015. Answering questions with complex semantic constraints on open knowledge bases. In *Proceedings of the 24th ACM International on Conference on Information and Knowledge Management*, pages 1301–1310.

Adams Wei Yu, David Dohan, Minh-Thang Luong, Rui Zhao, Kai Chen, Mohammad Norouzi, and Quoc V. Le. 2018. QANet: Combining Local Convolution with Global Self-Attention for Reading Comprehension. In *6th International Conference on Learning Representations (ICLR)*.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q Weinberger, and Yoav Artzi. 2019. Bertscore: Evaluating text generation with bert. *arXiv preprint arXiv:1904.09675*.

Yukun Zhu, Ryan Kiros, Rich Zemel, Ruslan Salakhutdinov, Raquel Urtasun, Antonio Torralba, and Sanja Fidler. 2015. Aligning books and movies: Towards story-like visual explanations by watching movies and reading books. In *The IEEE International Conference on Computer Vision (ICCV)*.

	Model	Quora			
		iBLEU	BLEU	ROUGE-1	ROUGE-2
Supervised	ResidualLSTM	12.67	17.57	59.22	32.40
	VAE-SVG-eq	15.17	20.04	59.98	33.30
	Pointer-generator	16.79	22.65	61.96	36.07
	Transformer	16.25	21.73	60.25	33.45
	+ Copy	17.98	24.77	63.34	37.31
	DNPG	**18.01**	**25.03**	**63.73**	**37.75**
Supervised (Wiki)	Pointer-generator	5.04	6.96	41.89	12.77
	Transformer + Copy	6.17	8.15	44.89	14.79
	Shallow fusion	6.04	7.95	44.87	14.79
	Multi-task learning	4.90	6.37	37.64	11.83
	+ Copy	7.22	9.83	47.08	19.03
	DNPG	**10.39**	**16.98**	**56.01**	**28.61**
Unsupervised	VAE	8.16	13.96	44.55	22.64
	CGMH	9.94	15.73	48.73	26.12
	UPSA	12.02	18.18	56.51	30.69
	PUP	14.91	19.68	59.77	30.47
	CorruptLM	12.08	16.80	49.13	26.15
	TA	**16.02**	**21.18**	**61.90**	**35.07**
	TA+SS	15.57	20.68	61.51	32.78
	TA+SS+DB	9.67	14.12	60.06	28.91
No model	**Copy-input**	**24.79**	**30.98**	**65.60**	**42.09**

Table 8: Automatic evaluation results on the QQP dataset. Models we (re)produced and SOTA results in each category are boldfaced. "Supervised (Wiki)" stands for models trained on WikiAnswers and evaluated on QQP.

	Model	Oracle Quality (10 sentences)			
		BLEU	ROUGE-1	ROUGE-2	ROUGE-L
Supervised	copy-input	18.4	54.4	27.2	49.2
	SCPN	21.3	53.2	30.3	51.0
	Transformer seq2seq	**32.8**	**63.1**	**41.4**	**63.3**
	+ diverse-decoding	24.8	56.8	33.2	56.4
	SOW-REAP (LSTM)	27.0	57.9	34.8	57.5
	SOW-REAP	30.9	62.3	40.2	61.7
Unsupervised	**CorruptLM (QQP)**	7.6	31.9	11.6	31.6
	TA+SS+DB (QQP)	22.0	**60.1**	**33.8**	**54.9**
	TA+SS+DB	**22.6**	59.8	33.2	54.5

Table 9: Automatic metrics results on the Para-NMT dataset. "(QQP)" stands for models finetuned on the non-parallel QQP dataset and evaluated on the ParaNMT dataset.

A Automatic Metric Results

We present automatic evaluation results on the previous metrics for QQP in Table 8 and for ParaNMT in Table 9. We can see that for QQP our task-adaptation model without Dynamic Blocking during inference achieves state-of-the-art results among unsupervised approaches. Had we based our judgments on Table 8, we would have mistakenly selected this one as our final model.

B Robustness to Grammar Errors

During the task-adaptation phase, our model in most cases has a grammatically correct sentence as the target sequence. Additionally, shuffling during that phase encourages the model to attend to the context during generation. These setups make our model reasonably robust to grammar errors so that it can paraphrase and normalize the input at the same time. Table 10 shows a case where we intentionally introduce grammar errors on subject-verb agreement, singular vs. plural, and verb inflections.

Input	Our approach are data-driven and can be apply across various situation.
Output	Our approach is data-driven and can be applied across various situations.
	Our approach is data-driven and can be applied across different situations.
	Our approach is data-driven and can be applied across diverse situations.
	Our approaches are data-driven and can be applied across various situations.
	Our data-driven approach can be applied across different situations.
	Our approaches are data-driven and can be applied across different situations.
	Our data-driven approach can be applied across diverse situations.
	Our approaches are data-driven and can be applied across diverse situations.

Table 10: Selected example of output candidates produced by our model where we intentionally introduce grammar errors (marked with underlines). We observe that all paraphrase candidates have these errors corrected.

We find that our model is in most cases robust to such errors. This trait is desired because we may face noisy inputs from users. Through early ablation studies, we observed that without shuffling during task-adaptation, the model was much less robust to grammar errors. Hence shuffling does more than just improving on the BERT-*i*BLEU metric (Table 6).

C Failure Modes

Though only occurring occasionally, our model exhibits multiple failure patterns. Hence we perform "anti-cherry-picking" and present in Table 11 some of such examples and the respective modes we outline. We hypothesize that the Antonym mode can be partially addressed by a lookup in the dictionary to additionally block the antonyms. Grammar errors are harder to resolve because they are usually apparent only after the whole sentence is generated. A grammar checker on the candidates may improve the situation. The swapping of subject and object shows that unsupervised approaches based on pretrained language models could only carry us so far till the syntactic-level. In its current form, it cannot handle semantic mistakes. For missing named entities, an NER tagger can help filter candidates that miss important entities. We leave addressing these failure modes as future work.

D Details of Dynamic Blocking

Block surface-form variations In our early experiments, we observed that when blocking a word (e.g. "give"), the model usually tries to generate its capitalized ("Give") or upper ("GIVE") version. From we human's perspective, these are usually not good paraphrases – intuitively we would prefer a different word. Similar to whole-word masking introduced in later versions of BERT,[19] we only

[19] https://github.com/google-research/bert

Failure mode	Input	Output
Antonym	How do I gain weight in a healthy way?	How do I lose weight in healthy ways?
Repeated words	What is the funniest movie to watch?	What is the most funniest film to see?
Grammar errors	Do spirits or ghosts exist?	Do ghost or spirit exist?
Subject ↔ object	How will you know you love someone?	How will you tell if someone loves you?
Missing named entity	A look of dismay came into luzhin's face.	A look of disappointment came into the face.

Table 11: Typical examples where our model fails to generate correct paraphrases. Words related to each failure mode are underlined.

block the beginning of the word rather than any subword.

Block Closed-Class Words We also leverage linguistic knowledge to help boost the quality of the paraphrases by avoiding blocking closed-class words, or functional words.[20] The closed classes in English include pronouns, determiners, conjunctions, and prepositions while open-class words correspond to nouns, lexical verbs, adjectives, and adverbs. There are two justifications for blocking these words. First, because they are closed-class, there are fewer synonyms available; second, blocking such words is error-prone. For example, changing determiners (e.g. from "*you*" to "*I*") may lead to syntactic or semantic errors, while modifying conjunctions (e.g. from "*and*" to "*or*") may lead to change in logical relationships.

Block Inflections In Section 5.2, we mentioned that BART can directly work with Dynamic Blocking without task-adaptation or self-supervision, but that results in lower quality, especially lacking syntactic variance because it is not trained with the shuffling strategy during task-adaptation. In addition, we found that without finetuning, BART tries to generate inflections of a word when it is blocked. To partially remedy this drawback, we use the *pattern* library[21] to enumerate all inflections of a word to block (e.g. for "*give*" we should also block "*gives*", "*gave*", "*giving*" and "*given*") in addition to all the other blocking schemes introduced in Section 3. This is available for most languages that involve inflections. We show in Table 12 the output candidates of a selected example with and without blocking inflections.

Retain Named Entities We also explore a variation of the system where we employ a separate Named Entity Recognition model to identify the named entities in the source sequence and prevent any tokens in these entities from appearing in the full block dictionary. This change ensures that all named entities are copied verbatim.

E Paraphrasing in German

We pair BART directly with Dynamic Blocking to generate paraphrases in German. In Table 13, we can see that all candidates (left column) have different surface forms, while all translations in English (right column)[22] share similar meanings.

F MTurk Instructions

To facilitate reproducibility, we include our MTurk instructions for the head-to-head and the Likert-based human studies (Figure 3 and 4). As mentioned in Section 3.5, we only provide guidelines on which paraphrases are better in general and leave the rest to the annotator's intuition.

[20] https://mailman.uib.no/public/corpora/attachments/20111124/6c58cb02/attachment.txt

[21] https://github.com/clips/pattern

[22] By Google Translator: https://translate.google.com/

Input		The random selection of pages must be performed by someone other than the player.
Output	Blocking inflections	The random choice of the pages must be performed by someone else than the player. The random selection of the pages must be performed by someone else than the user. The random selection of the pages must be executed by someone other than the user. The random collection of these pages must be performed by someone else than the player. The random selection of these pages must be executed by someone other than the user.
	No blocking inflections	The randomly selection of page must be perform by someone else than the players. The random choice of page must be performed by someone else than the player. The randomly selection of page must be perform by someone rather than the players. The random choice of page must be performed by someone rather than the player. The random collection of pages must be performed by someone else than the players.

Table 12: Selected example of output candidates produced by BART + Dynamic Blocking with and without blocking inflections. Grammar errors made by the latter due to wrong inflections are underlined.

	German	Translation from German
Input	Warum finden keine Brandschutzbelehrungen statt ?	Why are there no fire instructions?
Candidates	Warum lieen keine Geschutzbelehrungen statt?	Why were there no protection instructions?
	Warum finden keine Geschutzbelehrungen statt?	Why are there no protection instructions?
	Warum lieen keine Brandschutzbelehrungen statt?	Why weren't there any fire safety instructions?
	Warum finden keine Geschutzbelehrungen statt?	Why are there no protection instructions?
	Warum finden wir keine Brandschutzbelehrungen statt?	Why are we not giving fire safety instructions?

Table 13: Paraphrasing German input by directly applying Dynamic Blocking to BART. Translations on the right are given by the Google Translator, except that the first one is the ground-truth translation. Note that the candidates are ranked by multi-lingual BERT rather than RoBERTa-base which is only used to rank English outputs.

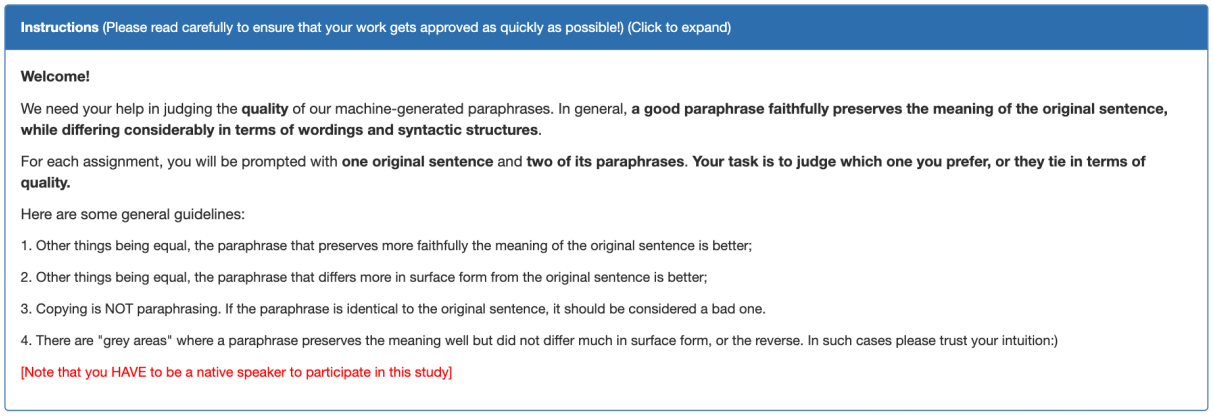

Figure 3: Interface of our MTurk studies for head-to-head comparisions with other models.

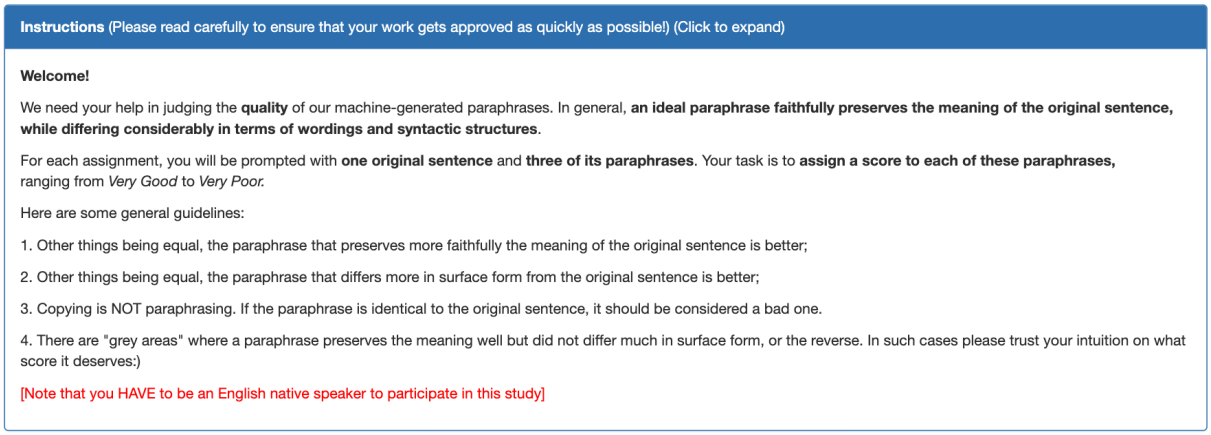

Figure 4: Interface of our MTurk studies for head-to-head comparisions with other models.