# REFLECTIVE DECODING: UNSUPERVISED PARAPHRASING AND ABDUCTIVE REASONING

**Anonymous authors**
Paper under double-blind review

## ABSTRACT

Pretrained Language Models (LMs) generate text with remarkable quality, novelty, and coherence. Yet applying LMs to the problems of paraphrasing and infilling currently requires direct supervision, since these tasks break the left-to-right generation setup of pretrained LMs. We present REFLECTIVE DECODING, a novel unsupervised approach to apply the capabilities of pretrained LMs to non-sequential tasks. Our approach is general and applicable to two distant tasks – *paraphrasing* and *abductive reasoning*. It requires no supervision or parallel corpora, only two pretrained language models: *forward* and *backward*. REFLECTIVE DECODING operates in two intuitive steps. In the *contextualization* step, we use LMs to generate many left and right contexts which collectively capture the meaning of the input sentence. Then, in the *reflection* step we decode in the semantic neighborhood of the input, conditioning on an ensemble of generated contexts with the reverse direction LM. We *reflect* through the generated contexts, effectively using them as an intermediate meaning representation to generate conditional output. Empirical results demonstrate that REFLECTIVE DECODING outperforms strong unsupervised baselines on both paraphrasing and abductive text infilling, significantly narrowing the gap between unsupervised and supervised methods. REFLECTIVE DECODING introduces the concept of using generated contexts to represent meaning, opening up new possibilities for unsupervised conditional text generation.

## 1 INTRODUCTION

Pretrained language models (LMs) have made remarkable progress in language generation. Trained over large amounts of unstructured text, models like GPT2 (Radford et al., 2019) leverage enhanced generation methods (Holtzman et al., 2020; Martins et al., 2020; Welleck et al., 2019) resulting in fluent and coherent continuations to given input text – e.g. news articles or stories.

However, it's unclear how to apply LMs to tasks that cannot be framed as left-to-right generation—e.g. paraphrasing and text infilling—without supervision. LMs undeniably model notions of "semantic neighborhood" and "contextual fit" inherent in these tasks: to predict the next sentence, a model must implicitly capture a subspace of similar sentences related to the given context. Can we leverage this implicit knowledge to apply pretrained LMs to non-sequential tasks without direct supervision?

We introduce REFLECTIVE DECODING—a novel decoding method that allows LMs to be applied to naturally distributional tasks like paraphrasing and text-infilling, without direct supervision. REFLECTIVE DECODING requires only two complementary LMs – one forward ($\overrightarrow{\text{LM}}$) and one backward ($\overleftarrow{\text{LM}}$). $\overrightarrow{\text{LM}}$ and $\overleftarrow{\text{LM}}$ are trained to generate text left-to-right (forward) and right-to-left (backward).

Inspired by the distributional hypothesis for representing the meaning of a word using other words it often co-occurs with (Firth, 1957), the two LMs generate *contexts* that collectively represent the meaning of a given sentence (the contextualization step). In the reflection step we decode with this meaning, by conditioning on an ensemble of these contexts with reverse-direction LMs.

Figure 1 (left) shows an example of REFLECTIVE DECODING applied to paraphrasing, with the left-side contexts omitted for clarity. Given an input $s_{src}$ : *How are circulatory system tissues formed?* the contextualization step generates contexts $c_i$ for $s_{src}$ with $\overrightarrow{\text{LM}}$, each capturing different aspects of the input sentence – e.g. $c_1$: *This is a medical question* situates the input as a question, and $c_2$: *As*

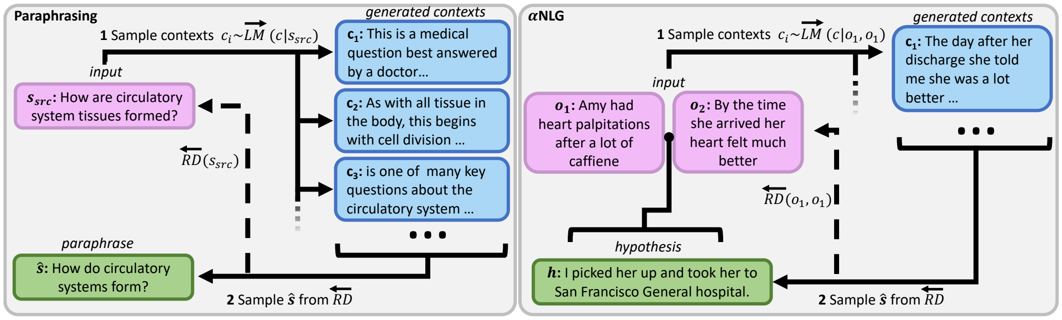



Figure 1: An illustration of how REFLECTIVE DECODING is applied to paraphrasing and $\alpha$NLG. **Only** the right-context is shown, although **both** are used in practice. First (1) the contextualization step captures the meaning of an input by generating many contexts for it. Then, (2) the reflection step samples generations using this meaning with $\overleftarrow{\text{RD}}$ (the REFLECTIVE DECODING sampling function). $\overleftarrow{\text{RD}}$ uses the reverse-direction language model $\overleftarrow{\text{LM}}$ to sample in the semantic neighborhood of the input, with an ensemble of contexts that should also be likely to sample input (dashed arrow).

*with all tissue in the body...* presents an elaboration of the central concept in $s_{src}$ (tissue formation). Collectively, many contexts capture the meaning of the input sentence. Next, we sample outputs in the reflection step. By conditioning on the generated contexts with a backwards language model ($\overleftarrow{\text{LM}}$) in a weighted ensemble $\overleftarrow{\text{RD}}$, we *reflect* back from the contexts to generate a sentence with the same meaning as the input text.

REFLECTIVE DECODING shows strong unsupervised: On the Quora paraphrasing dataset, we test with multiple levels of $Novelty$ (variation from the source sentence) finding one setting ($\text{RD}_{30}$) outperforms unsupervised baselines on all but one metric, and supervised baselines on both the SARI metric and human evaluation. Applying REFLECTIVE DECODING to $\alpha$NLG (Bhagavatula et al., 2020)—a text infilling task—we outperform the unsupervised baseline on overall quality by 30.7 points, significantly closing the gap with supervised methods. In both applications, REFLECTIVE DECODING proceeds without domain finetuning, directly using pretrained models.

Empirical results suggest the possibility that completely unsupervised generation can solve a number of tasks with thoughtful decoding choices, analogous to GPT-3 (Brown et al., 2020) showing the same for thoughtfully designed contexts. We provide an intuitive interpretation of REFLECTIVE DECODING in §2.7: sampling while prioritizing contextual (i.e. distributional) similarity with respect to the source text. REFLECTIVE DECODING demonstrates how far unsupervised learning can take us, when we design methods for eliciting specific kinds of information from pretrained LMs.

## 2 METHOD

### 2.1 NOTATION

We begin by defining notation used in explaining our method. Arrows are used to indicate the order in which a sampling function conditions on and generates tokens: $\rightarrow$ indicates generating from the left-most token and proceeding to the right, while $\leftarrow$ indicates going from the rightmost token to the left. One example of this is applying these to Language Models: $\overrightarrow{\text{LM}}$, commonly referred to as a "forward" LM processes and generates tokens from left to right. Any token generated by $\overrightarrow{\text{LM}}$ is always directly following, or to the right of context $c$. In contrast, $\overleftarrow{\text{LM}}$ is what is typically called a "backwards" LM, and proceeds left from the right-most token. When a token is generated, it is to the left of the input context $c$.

This holds true for other functions. When we refer to the sampling function of our method (RD), its arrow indicates whether it begins by generating the left-most or right-most token of the output sentence ($\overrightarrow{\text{RD}}$ or $\overleftarrow{\text{RD}}$, respectively). This also implicitly indicates which context it is conditioning on (see §2 for more details): $\overrightarrow{\text{RD}}$ conditions on left context, and extends it in the left-to-right direction to

**Algorithm 1:** Learn REFLECTIVE DECODING sampling function (right-to-left)

**Input:** Forward language model LM, backward language model LM, Source text: $s_{src}$

1: Sample contexts, $c_1...c_{n_c}$ $\mathrm{LM}(cjs_{src})$
2: Initialize parameters $\mathbf{w} = w_1...w_{n_c}$ s.t. $\sum w_i = 1, w_i \quad 0$
3: $\mathrm{RD}(s)$ / $\prod_i \mathrm{LM}(sjc_i)^{w_i}$ normalized by token (equation **??**)
4: learn $\mathbf{w}$ = $\arg\max_{\mathbf{w}} \mathrm{RD}(s_{src})$ s.t. $\sum w_i = 1, w_i \quad 0$

**Output:** RD

generate the output. $\overleftarrow{\mathrm{RD}}$ conditions on the right context, and generates backwards (right-to-left) to extend it into the desired output.

### 2.2 OVERVIEW

REFLECTIVE DECODING is an unsupervised generation method that conditions on the content of an input text, while abstracting away its surface form. This is useful in paraphrasing where we want to generate a new surface form with the same content, but also when the surface form is difficult to decode from. In text infilling for instance, unidirectional LMs cannot condition on bidirectional context, but REFLECTIVE DECODING avoids directly conditioning on surface form, generating contexts that capture the desired meaning in aggregate.

To justify how generated contexts do this, consider the example from figure 1 with input $s_{src}$ : *How are circulatory system tissues formed?* By generating contexts for $s_{src}$, we capture different aspects: $c_1$ situates $s_{src}$ as a question (*This is a medical question...*), while $c_2$ and $c_3$ explore central concepts (*as with all tissue...*; *about the circulatory system*). While each context could follow many sentences, together they form a fingerprint for $s_{src}$. A sentence that could be followed by all of $c_1, c_2, c_3$ will likely be a question ($c_1$) about tissue formation ($c_2$) and the circulatory system ($c_3$), semantically similar to $s_{src}$ or even a paraphrase ($\hat{s}$: *How do circulatory systems form?*).

REFLECTIVE DECODING works on this principle. For an input text $s_{src}$, we use a language model to generate contexts that serve as a fingerprint for meaning. This is the contextualization step. Then in the reflection step, we sample generations that also match these contexts. We consider right contexts generated by $\overrightarrow{\mathrm{LM}}$ here, but both directions are used. To effectively and efficiently sample with content in $s_{src}$, we learn the REFLECTIVE DECODING sampling function:

$$\overleftarrow{\mathrm{RD}}\,(\mathrm{s}) = \frac{\prod_i \overleftarrow{\mathrm{LM}}(s|c_i)^{w_i}}{Z(s, \mathbf{c}, \mathbf{w})} \tag{1}$$

This can be understood as a Product of Experts model (Hinton, 2002) between language model distributions ($\overleftarrow{\mathrm{LM}}$) with different contexts, where the $Z$ function just normalizes for token-by-token generation (see equation 2 for its definition, with arguments source s, contexts **c**, weights **w**). This matches the intuition above, that a paraphrase should fit the same contexts as the source: $\overleftarrow{\mathrm{LM}}$ conditions on an ensemble of these contexts, further informed by weights $w_i$ that maximize the probability of $s_{src}$ under $\overleftarrow{\mathrm{LM}}$. In effect, we use weight contexts $c_i$ to best describe the source.

$\overleftarrow{\mathrm{RD}}$ samples generations that would also have generated these contexts, as in the example above. This can be seen as sampling to minimize a notion of contextual difference or maximize similarity between the sampled text $s$ and $s_{src}$ (§2.7).

REFLECTIVE DECODING returns samples in the semantic neighborhood of $s_{src}$, but the specific application directs how these are ranked. In paraphrasing, we want the semantically closest sample, using a contextual score (equation 3). In text infilling ($\alpha$NLG) the goal is to fill-in the narrative "gap" in the surrounding text rather than maximize similarity (equation 4).

| Task: Paraphrasing | |
|---|---|
| what is it like to have a midlife crisis? | |
| $RD_{30}$ | what does it mean to have a midlife crisis? |
| $RD_{45}$ | what do you do when you have a midlife crisis? |
| is it possible to make money as a film critic? | |
| $RD_{30}$ | is there a way to make money as a film critic? |
| $RD_{45}$ | is it possible to make a living as a movie critic? |

| Task: $\alpha$NLG | |
|---|---|
| $o_1$: Ray hung a tire on a rope to make his daughter a swing. __?__ $o_2$: Ray ran to his daughter to make sure she was okay. | |
| RD | He put her on the swing, and while she was on the swing, she fell off and was lying on the ground. |
| $o_1$: Tom and his family were camping in a yurt. __?__ $o_2$: He chased it around until it left the yurt. | |
| RD | He went to the yurt and found a bear that was in the yurt |

Figure 2: Example generations of REFLECTIVE DECODING on paraphrasing and abductive text infilling ($\alpha$NLG). $RD_{45}$ encourages more difference from the input than $RD_{30}$ (§3.1).

## 2.3 REFLECTIVE DECODING

Here, we explicitly describe the steps required to generate with REFLECTIVE DECODING. Centrally, we construct a sampling function RD. We describe right-to-left $\overleftarrow{\mathrm{RD}}$ in algorithm 1 but also use left-to-right $\overrightarrow{\mathrm{RD}}$ in practice (symmetrically described in §B.1 by reversing LMs). Algorithm 1 proceeds using only the input $s_{src}$, and two LMs (forward $\overleftarrow{\mathrm{LM}}$ and backward $\overrightarrow{\mathrm{LM}}$). We explain algorithm 1 below:

**contextualization step** (line 1) We generate right contexts $c_i$ that follow the source text $s_{src}$, using forward language model $\overrightarrow{\mathrm{LM}}$. Following §2.2 and figure 1 these represent in meaning in $s_{src}$.

**reflection step** (lines 2-4) Next, we define the sampling function $\overleftarrow{\mathrm{RD}}$ we will use to generate outputs. As discussed in §2.2, this takes the form of a Product of Experts model normalized by token (equation 1). More explicitly:

$$\overleftarrow{\mathrm{RD}}\,(\mathrm{s}) = \frac{\prod_i \overleftarrow{\mathrm{LM}}(s|c_i)^{w_i}}{\prod_{j=0}^{jsj} \sum_{t2V} \prod_i \overleftarrow{\mathrm{LM}}(t|s_{j+1:jsj} + c_i)^{w_i}} \tag{2}$$

Algorithmically, the main step is learning informative weights $w_i$ for the generated contexts. As outlined in §2.7, we would like to sample sentences that fit the context of $s_{src}$. Intuitively, $s_{src}$ best fits its own context, and so we learn weights $w_i$ to maximize probability of sampling $s_{src}$.

we initialize (line 2) and learn (line 4) weights that maximize the probability of generating $s_{src}$ under the sampling function $\overleftarrow{\mathrm{RD}}$ (equation **??**). From §2.7, §A.1 we are sampling text with low "contextual distance" from $s_{src}$; this guides weight-learning and implies weights form a proper distribution (line 2,4).

Finally, in the reflection step we use $\overleftarrow{\mathrm{RD}}$ to sample text conditioned on the meaning of $s_{src}$, captured by the generated, weighted contexts (applied in §2.5 and §2.6). $\overleftarrow{\mathrm{RD}}$ samples right-to-left, and a similar left-to-right sampling function $\overrightarrow{\mathrm{RD}}$ is learned symmetrically by reversing the roles of $\overrightarrow{\mathrm{LM}}$ and $\overleftarrow{\mathrm{LM}}$ (detailed in §B.1). We describe some practical aspects for this process in §2.4.

## 2.4 IMPLEMENTATION

Here, we cover implementation details for §2.3.

**Weight Pruning** In practice, we sample tens of contexts (line 1), many ending up with negligible weight in the final sampling function. For efficiency in sampling from an ensemble of these contexts (equation 2), we then drop all but the top $k_c$ contexts and renormalize weights. Thus, $k_c < n_c$ is the actual number of contexts used during the reflection step of §2.3.

**Parameters** In line 1, we sample $n_c$ contexts to describe the source $s_{src}$. We use nucleus sampling (Holtzman et al., 2020) (described in §5) with parameter $p_c$, and a maximum length of $len_c$. As stated in **Weight Pruning**, we drop all but the top $k_c$ contexts by weight. Once a REFLECTIVE DECODING sampling function is learned, we sample $n_{\hat{s}}$ generations, of length $len_{\hat{s}}$. Again, we use nucleus sampling with $p$ picked by entropy calibration (§B.3). Values for all parameters are available in §B.4.

**Language Models** We train large forward ($\overrightarrow{\mathrm{LM}}$) and backward ($\overleftarrow{\mathrm{LM}}$) language models based on GPT2 (Radford et al., 2019) using the OpenWebText training corpus (Gokaslan & Cohen, 2019). Our implementation details follow those of past work retraining GPT2[1] (Zellers et al., 2019).

### 2.5 APPLICATION: PARAPHRASING

Following §2.3 the REFLECTIVE DECODING sampling function is learned in each direction ($\overrightarrow{\mathrm{RD}}$, $\overleftarrow{\mathrm{RD}}$) using the input sentence $s_{src}$. Then, $n_{\hat{s}}$ generations are sampled from both $\overrightarrow{\mathrm{RD}}$ and $\overleftarrow{\mathrm{RD}}$:

$$\hat{s}_1, ..., \hat{s}_{n_{\hat{s}}} \sim \overrightarrow{\mathrm{RD}}, \hat{s}_{n_{\hat{s}}+1}, ..., \hat{s}_{2\ \ n_{\hat{s}}} \sim \overleftarrow{\mathrm{RD}}$$

This gives a robust set of candidates using both sides of context. They are in the semantic neighborhood of $s_{src}$ but must be ranked. REFLECTIVE DECODING is based on a notion of similarity centered on contextual distance posed as cross-entropy (equation 6 and §2.7), so we use this as a final scoring function leveraging the generated contexts of $\overrightarrow{\mathrm{RD}}$ and $\overleftarrow{\mathrm{RD}}$:

$$score(\hat{s}) = \frac{1}{n_c} \sum_{c_{rh}} \overrightarrow{\mathrm{LM}}(c_{rh}|\hat{s}) + \frac{1}{n_c} \sum_{c_{lh}} \overleftarrow{\mathrm{LM}}(c_{lh}|\hat{s}) \tag{3}$$

Where $c_{rh}$ are the generated contexts used in $\overleftarrow{\mathrm{RD}}$, and $c_{lh}$ for $\overrightarrow{\mathrm{RD}}$. Intuitively, we see this as how well $\hat{s}$ fits the contexts of $s_{src}$, estimated with finite samples on each side.

### 2.6 APPLICATION: ABDUCTIVE REASONING

Abductive natural language generation ($\alpha$NLG from Bhagavatula et al. (2020)) is the task of filling in the blank between 2 observations $o1$ and $o2$, with a hypothesis $h$ that abductively explains them. Approaching this problem unsupervised is challenging, particularly with unidirectional language models which cannot naturally condition on both sides when generating $h$.

REFLECTIVE DECODING simplifies this problem. Using concatenated $o_1 + o_2$ as $s_{src}$ in algorithm 1, we learn a REFLECTIVE DECODING sampling function that captures the content of both observations. We are interested in sampling in between $o1$ and $o2$, so when sampling hypotheses $h$ from $\overleftarrow{\mathrm{RD}}$ we condition on the right-side observation $o_2$ (and vice-versa for $\overrightarrow{\mathrm{RD}}$ and $o_1$):

$$h_1, ..., h_{n_{\hat{s}}} \sim \overleftarrow{\mathrm{RD}}(h|o_2),\ \ h_{n_{\hat{s}}+1}, ..., h_{2\ \ n_{\hat{s}}} \sim \overrightarrow{\mathrm{RD}}(h|o_1)$$

Note that both $\overrightarrow{\mathrm{RD}}$ and $\overleftarrow{\mathrm{RD}}$ each contain information about **both** $o_1$ and $o_2$. Here we have a different goal than the paraphrasing application: we would like to explain the gap between $o_1$ and $o_2$, rather than rephrase $o_1 + o_2$ into a new surface form. $\overrightarrow{\mathrm{RD}}$ and $\overleftarrow{\mathrm{RD}}$ sample semantically related sentences to the input, and so we simply sample with higher diversity (higher $p$ in Nucleus Sampling) than for paraphrasing, to encourage novel content while still using the information from $o1 + o2$.

We also use a task-specific scoring function to rank sampled hypotheses. We would like a hypothesis that best explains both observations, and so use language models to measure this:

$$score(h) = \overleftarrow{\mathrm{LM}}(o1|h + o2) + \overrightarrow{\mathrm{LM}}(o2|o1 + h) \tag{4}$$

Adding $h$ should help explain each observation given the other, meaning $o2$ is follows from $o1 + h$ and $o1$ from $h + o2$. To filter hypotheses that only explain one of the two observations, we remove any that make either observation less explained than no hypothesis, imposing:

$\mathrm{LM}(o_1 j h + o_2) > \mathrm{LM}(o_1 j o_2), \mathrm{LM}(o_2 j o_1 + h) > \mathrm{LM}(o_2 j o_1)$

[1]https://github.com/yet-another-account/openwebtext

### 2.7 Intuitions and Theory

Here, we motivate and derive Reflective Decoding as a way to sample generations under a notion of contextual "fit" with a source text, deriving the sampling function of equation **??**. We start by considering how to compare a generation $\hat{s}$ with input $s_{src}$.

We follow a distributional intuition (Firth, 1957), that textual meaning can be understood by the contexts in which text appears. Many distributional approaches learn contentful neural representations by predicting context given input text (Mikolov et al., 2013; Kiros et al., 2015), then compare these representations for meaning. Instead, we compare contexts directly. Specifically, judging the difference in meaning between texts $s_{src}$ and $\hat{s}$ by their divergence:

$$D_{KL}(\overrightarrow{\text{LM}}(c|s_{src}), \overrightarrow{\text{LM}}(c|\hat{s})) \tag{5}$$

For simplicity, we use $\overrightarrow{\text{LM}}$ to denote both the theoretical left-to-right distribution of text, and the model distribution estimating it. $\overrightarrow{\text{LM}}(c|s)$ is the distribution over right contexts $c$ given sentence s, so equation 5 can be understood as how different the right-contexts we expect $s_{src}$ and $\hat{s}$ to appear in are. Note, while we use right-hand context here, this explanation symmetrically applies to left-hand.

Measuring $D_{KL}$ exactly is infeasible, but for generation we are mainly interested in ranking or optimizing for this score (e.g. picking the best paraphrase $\hat{s}$). We take inspiration from language models, using a finite sample estimate of cross-entropy as an effective proxy for $D_{KL}$:

$$\hat{H}(\overrightarrow{\text{LM}}(c|s_{src}), \overrightarrow{\text{LM}}(c|\hat{s})) = \frac{1}{N} \sum_{c_i \sim \overrightarrow{\text{LM}}(c|s_{src})} -\log \overrightarrow{\text{LM}}(c_i|\hat{s}) \tag{6}$$

Where $c_i \sim \overrightarrow{\text{LM}}(c|s_{src})$ indicates contexts sampled from the LM conditioned on the input $s_{src}$. This objective makes intuitive sense: we want similar sentence $\hat{s}$ to rank highly, so we "imagine" contexts for $s_{src}$ and choose $\hat{s}$ that most generates these contexts. Optimal $\hat{s}$ fills approximately the same contextual hole as $s_{src}$, minimizing this "contextual distance".

In this form, $\hat{H}$ requires a fully generated $\hat{s}$ to compare, although we are trying to generate $\hat{s}$ for which this is low. We leverage the symmetric nature of the relationship between text and context to "reflect" equation 6, into a function from which we can sample:

$$\overleftarrow{\text{RD}}(\hat{s}_j|, \hat{s}_{j+1:n}) = \frac{\prod_i \overleftarrow{\text{LM}}(\hat{s}_j|\hat{s}_{j+1:n} + c_i)^{w_i}}{\sum_{t \in V} \prod_i \overleftarrow{\text{LM}}(t|\hat{s}_{j+1:n} + c_i)^{w_i}} \tag{7}$$

(equivalent to equation **??**, derived in §A.1) $\hat{s}_j$ is the $j^{th}$ token in $\hat{s}$ (sampled right-to-left from n to 0), and $V$ is vocabulary. Weights $w_i$ are learned, aligning probability with contextual similarity to $s_{src}$ by maximizing probability of $s_{src}$ (best fits its own context). In effect, $\hat{s}$ with low contextual distance with source $s_{src}$ is likely. We can use left or right context by reversing the role of the LMs.

## 3 Experiments

### 3.1 Task: Paraphrase Generation

**Task:** Following past work, we test our paraphrasing method (§2.5) on the Quora question pair dataset. We hold out 1000 examples for testing, with the rest for training and validation (used by supervised baselines), disallowing overlap with the test set.

**Metrics:** Following past work, we include automatic metrics BLEU (Papineni et al., 2002), METEOR (Denkowski & Lavie, 2014), and TER$_p$ (Snover et al., 2009). These measure agreement with references, but high overlap between references and inputs means copying input as-is gives high scores (Mao & Lee, 2019); copying source sentences as-is beats all models on these metrics (table 1).

Past work has emphasized the important challenge of offering a novel phrasing in this task (Liu et al., 2010; Chen & Dolan, 2011) beyond simply agreeing in meaning. Reference-agreement metrics don't explicitly measure this novelty. We address this in 3 ways. First, we explicitly quantify a simple notion of novelty:

$$Novelty(\hat{s}) = 100 - BLEU(\hat{s}, s_{src}) \tag{8}$$

| | Method | SARI″ | BLEU″ | METEOR″ | $\text{TER}_P$# | Human″ | $Novelty$ ″ |
|---|---|---|---|---|---|---|---|
| *Human* | Source | 17.8 | 56.0 | 37.6 | 48.0 | - | 0.0 |
| | Reference | 91.9 | 100.0 | 100.0 | 0.0 | 71.7 | 43.9 |
| *Supervised* | PG-IL | 32.8 | **49.1** | 33.8 | **49.0***| 29.4 | 24.4 |
| | DiPS | **38.8** | 41.0 | 27.9 | 56.0 | 36.6 | **48.5*** |
| | BART | 36.1 | 44.7 | **34.7*** | 66.0 | 46.1 | 35.2 |
| *Supervised (Bilingual)* | MT | 35.6 | 48.1 | 33.5 | 52.0 | 59.3 | 26.8 |
| *Unsupervised* | R-VQVAE | 27.2 | 43.6 | 32.3 | 60.0 | 33.5 | 26.2 |
| | $\text{CGMH}_{Top}$ | 32.3 | 42.0 | 28.2 | 59.0 | 27.0 | 27.6 |
| | $\text{CGMH}_{30}$ | 33.9 | 40.9 | 27.5 | 60.0 | 31.5 | 29.7 |
| | $\text{CGMH}_{45}$ | 32.6 | 33.8 | 23.4 | 65.0 | 15.8 | 44.5 |
| | $\text{RD}_{Top}$ (Us) | 29.0 | **49.9*** | **33.9** | **52.0** | 27.5 | 20.8 |
| | $\text{RD}_{30}$ (Us) | **40.0*** | 46.8 | 32.2 | 57.0 | 63.2 | 30.0 |
| | $\text{RD}_{45}$ (Us) | 38.6 | 39.9 | 28.9 | 65.0 | 61.1 | **63.1** |

Table 1: Model performance on the Quora test split. **Bold** indicates best for model-type, * indicates best overall (excluding human). The first 5 columns are measures of quality, while the last measures novelty (equation 8) or difference from input. We rerun evaluations from past work.

to measure how agreement with the reference trades off with repeating the input. Second, we include the SARI metric (Xu et al., 2016) which explicitly balances novelty from input with reference overlap. Third, we quantify an overall human quality metric: the rate at which annotators find paraphrases fluent, consistent with input meaning, and novel in phrasing. This is the "Human" column in table 1.

3 annotators evaluate models on 204 inputs for fluency, consistency, and novelty on Amazon Mechanical Turk. The "Human" metric is the rate that examples meet the threshold for all 3: fluent enough to understand, with at most minor differences in meaning and at least minor differences in wording. We find rater agreement with Fleiss' $\kappa$ (Fleiss, 1971) to be 0.40 (fluency threshold), 0.54 (consistency threshold), 0.77 (novelty threshold) and 0.48 (meets all thresholds) indicating moderate to substantial agreement (Landis & Koch, 1977). Human evaluation is described more in §C.2.

**Baselines:** Parameters for REFLECTIVE DECODING are given in §B.4. We mainly compare against 2 unsupervised baselines: Controlled Sentence Generation by Metropolis Hastings (CGMH from Miao et al. (2019)), and the residual VQ-VAE of Roy & Grangier (2019b) (R-VQVAE). This is a cross-section of recent approaches (VAE, editing).

We also compare against a machine-translation approach (see Sec 5), by pivoting through German using Transformer (Vaswani et al., 2017) models trained on WMT19 data (Barrault et al., 2019). MT has access to bilingual data, and many past unsupervised paraphrasing works do not compare against it. Thus, we include it in a separate section in our results, Table 1.

We include supervised baselines: the pointer generator trained by imitation learning (PG-IL) as in Du & Ji (2019), the diversity-promoting DiPS model (Kumar et al., 2019), and a finetuned BART (Lewis et al., 2019) model, which uses a more complex pretraining method than our LMs. Note that DiPS generates multiple diverse paraphrases so we pick one at random.

CGMH and REFLECTIVE DECODING both return multiple ranked paraphrases. We can easily control for $Novelty$ by taking the highest-ranked output that meets a $Novelty$ threshold. For each, we have a version with no threshold ($Top$), and with thresholds such that average $Novelty$ is 30 and 45.

### 3.2 TASK: ABDUCTIVE NLG

**Task:** The Abductive natural language generation task ($\alpha$NLG) presented in Bhagavatula et al. (2020) requires generating a hypothesis that fits between observations $o_1$ and $o_2$, and explains them. We apply REFLECTIVE DECODING to this problem as outlined in §2.6, using available data splits.

**Baselines:** Parameters for REFLECTIVE DECODING are given in §B.4. We include baselines from the original work: different supervised variants of GPT2 large with access to the observations, and optionally COMET (Bosselut et al., 2019) embeddings or generations. We include an unsupervised baseline of GPT2 conditioned on $o_1$ + $o_2$ directly.

**Metrics:** For human evaluation, over 1000 examples we ask 3 raters on Amazon Mechanical Turk about agreement between h and $o_1$, $o_2$, both, and overall quality on 4-value likert scales. We found Fleiss' kappa (Fleiss, 1971) of 0.31, 0.29, 0.28, and 0.30 respectively, indicating fair agreement (Landis & Koch, 1977).

## 4 RESULTS AND DISCUSSION

**Paraphrasing:** On automatic metrics from past works (BLEU, METEOR, TER$_P$) our lowest-$Novelty$ model setting (RD$_{Top}$) achieves the highest unsupervised scores, and highest overall on BLEU. Other high scoring rows (Source, PG-IL) have similarly low-$Novelty$ outputs . The SARI metric explicitly balances novelty (i.e. difference from the source) with correctness (i.e. similarity to the reference). On SARI we see such low-$Novelty$ models perform worse. The best overall model on SARI is our medium-$Novelty$ setting (RD$_{30}$) which outperforms MT and supervised models.

Ultimately, human annotation is the only way to validate the results of other metrics. Our human evaluation measures what fraction of outputs are found to be fluent, consistent, and novel. As with SARI, both our mid and high-$Novelty$ models perform quite well. Our medium-$Novelty$ setting RD$_{30}$ achieves the highest score overall (63.1), slightly higher than RD$_{45}$ (61.1) and MT (59.3). Further, our human evaluation validates SARI as a reasonable proxy, as they share the same top-5 models.

REFLECTIVE DECODING is able to compete on previously used quality metrics that favor low-$Novelty$, but can easily produce more varied outputs preferred by humans. RD$_{45}$ exceeds the novelty of even the human reference, but is still among the highest ranked models by SARI and Human.

**$\alpha$NLG:** Results on $\alpha$NLG (table 2) present a strong case that REFLECTIVE DECODING can effectively use bidirectional context. Strong hypotheses use information from both the initial observation $o_1$ and the future observation $o_2$. Humans ranked the ability of REFLECTIVE DECODING to capture this 0.44, about 25 points above the unsupervised baseline and only 10 points below the best supervised method tested. We see similar results for overall evaluation.

We also include example generations in figure 2 to demonstrate the ability of REFLECTIVE DECODING to combine $o_1$ and $o_2$. For example, *He put her on the swing, and while she was on the swing, she fell off and was lying on the ground.* incorporates information from both observations. Specifically, it takes into account the swing that Ray is building for his daughter which is only mentioned in $o_1$, and hypothesises about a potential injury due to Ray checking on his daughter in $o_2$.

Overall, the strong performance of REFLECTIVE DECODING on $\alpha$NLG verifies applications beyond paraphrasing. Indeed, all that was required to apply REFLECTIVE DECODING was a source text whose content the generation should adhere to ($o_1$ and $o_2$ in this case), and two unidirectional LMs.

## 5 RELATED WORK

**Decoding Techniques** Holtzman et al. (2020) present nucleus sampling, a decoding technique to improve generation quality. It utilized distribution truncation at the token level. When generating a token at position i (given context $x_{1:i-1}$), nucleus sampling operates by keeping the smallest vocabulary $V_p$ set that satisfies:

$$\sum_{x \in V_p} P(x|x_{1:i-1}) \geq p$$

where $p$ is the sampling parameter and $P$ is the generating distribution, which is then renormalized to the reduced vocabulary. Rather than methods like tok-k sampling which take a static number of most-likely samples, nucleus sampling takes a static segment of the probability mass, $p$. Nucleus sampling is orthogonal to REFLECTIVE DECODING, which instead extends LM decoding to a new set of problems and in fact includes nucleus sampling as a subroutine. Kajiwara (2019) use a constrained decoding scheme to improve paraphrasing, but require a supervised system.

**Distributional Intuitions** A key aspect of REFLECTIVE DECODING is using a distributional intuition to represent the meaning of a text through many contexts. Kiros et al. (2015); Miao et al. (2019) quantify semantic and Lin & Pantel (2001) identify paraphrastic relationships under similar intuitions. A major point of difference between past work and ours is that we generate explicit contexts to represent meaning, allowing unsupervised generation back from representations, while past work typically attempts to compress the full contextual distribution into a fixed-length vector.

**Unsupervised Paraphrasing** One approach trains neural variational auto-encoders unsupervised to represent source sentences, then decodes from these representations to paraphrase (Roy & Grangier, 2019a; Bao et al., 2019). This requires training specialized representations, whereas REFLECTIVE DECODING applies general-purpose LMs. We compare against Roy & Grangier (2019a).

Paraphrasing by editing the input (Miao et al., 2019; Liu et al., 2019) has shown promise. Like REFLECTIVE DECODING, these approaches can be applied without training specialized models , but are necessarily limited by edit-paths and local minima, as edits are often restricted to single-word replacement, insertion, and deletion.

| | $o_1$ | $o_2$ | $o_1 + o_2$ | all |
|---|---|---|---|---|
| human | 86.7 | 84.3 | 78.0 | 83.8 |
| *Supervised* | | | | |
| COMeT$_{\text{Emb}}$+GPT2 | 72.4 | **61.9** | 55.1 | 60.1 |
| COMeT$_{\text{Txt}}$+GPT2 | 72.1 | 60.3 | 54.6 | 59.4 |
| $O_1$-$O_2$-Only | **72.5** | 61.6 | **55.7** | **60.9** |
| *Unsupervised* | | | | |
| GPT2-Fixed | 24.2 | 21.0 | 18.5 | 19.3 |
| Reflect Decoding | **56.7** | **55.3** | **46.2** | **50.0** |

Table 2: Model performance on $\alpha$NLG. The first 3 scores query agreement between hypothesis and given observation(s), and "all" indicates overall judgement.

REFLECTIVE DECODING and MT-bases paraphrasing both pivot through an alternative textual form to paraphrase (context and translation, resp.). But MT paraphrase systems cycle-translate through a pivot language (Federmann et al., 2019; Wieting & Gimpel, 2018), which requires supervised bilingual translation data, with an implicit notion of cross-lingual paraphrasing.

**Novelty in Paraphrasing** Mao & Lee (2019) observe that paraphrases close to the source often win on automatic quality metrics. However, dissimilarity from the source seems to correlate with human notions of paraphrasing (Liu et al., 2010), necessitating more nuanced metrics. Alternative metrics that consider novelty alongside quality have previously been used (Sun & Zhou, 2012; Federmann et al., 2019). The SARI metric (Xu et al., 2016), included here, combines these notions into a single metric. Kumar et al. (2019) increase novelty through their diversity-promoting sampling method.

# 6 CONCLUSIONS

We present REFLECTIVE DECODING, a novel unsupervised text generation method for tasks that do not fit the left-to-right generation paradigm. REFLECTIVE DECODING uses two language models to generate contexts that collectively represent the meaning of input text. It significantly outperforms unsupervised baselines in quality and diversity for paraphrasing. Further, in abductive natural language generation it outperforms the unsupervised baseline by a wide margin and closes the gap with supervised models. REFLECTIVE DECODING introduces the concept of using generated contexts to represent meaning, opening up new possibilities for unsupervised conditional text generation.

# REFERENCES

Yu Bao, Hao Zhou, Shujian Huang, Lei Li, Lili Mou, Olga Vechtomova, Xinyu Dai, and Jiajun Chen. Generating sentences from disentangled syntactic and semantic spaces. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pp. 6008–6019, 2019.

Loïc Barrault, Ondřej Bojar, Marta R Costa-Jussà, Christian Federmann, Mark Fishel, Yvette Graham, Barry Haddow, Matthias Huck, Philipp Koehn, Shervin Malmasi, et al. Findings of the 2019 conference on machine translation (wmt19). In *Proceedings of the Fourth Conference* *on Machine Translation (Volume 2: Shared Task Papers, Day 1)*, pp. 1–61, 2019. URL http://www.aclweb.org/anthology/W19-5301.

Chandra Bhagavatula, Ronan Le Bras, Chaitanya Malaviya, Keisuke Sakaguchi, Ari Holtzman, Hannah Rashkin, Doug Downey, Scott Yih, and Yejin Choi. Abductive commonsense reasoning. *ICLR*, 2020.

Antoine Bosselut, Hannah Rashkin, Maarten Sap, Chaitanya Malaviya, Asli Celikyilmaz, and Yejin Choi. Comet: Commonsense transformers for automatic knowledge graph construction. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pp. 4762–4779, 2019.

Tom B Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *arXiv preprint arXiv:2005.14165*, 2020.

David Chen and William Dolan. Collecting highly parallel data for paraphrase evaluation. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pp. 190–200, Portland, Oregon, USA, June 2011. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/P11-1020.

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

Wanyu Du and Yangfeng Ji. An empirical comparison on imitation learning and reinforcement learning for paraphrase generation. In *EMNLP/IJCNLP*, 2019.

Christian Federmann, Oussama Elachqar, and Chris Quirk. Multilingual whispers: Generating paraphrases with translation. In *Proceedings of the 5th Workshop on Noisy User-generated Text (W-NUT 2019)*, pp. 17–26, Hong Kong, China, November 2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-5503. URL https://www.aclweb.org/anthology/D19-5503.

John R Firth. A synopsis of linguistic theory, 1930-1955. *Studies in linguistic analysis*, 1957.

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

Aaron Gokaslan and Vanya Cohen. Openwebtext corpus, 2019. URL http://Skylion007.github.io/OpenWebTextCorpus.

Geoffrey E Hinton. Training products of experts by minimizing contrastive divergence. *Neural computation*, 14(8):1771–1800, 2002.

Ari Holtzman, Jan Buys, Maxwell Forbes, and Yejin Choi. The curious case of neural text degeneration. *ICLR*, 2020.

Tomoyuki Kajiwara. Negative lexically constrained decoding for paraphrase generation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pp. 6047–6052, 2019.

Maurice G Kendall. A new measure of rank correlation. *Biometrika*, 30(1/2):81–93, 1938.

Ryan Kiros, Yukun Zhu, Russ R Salakhutdinov, Richard Zemel, Raquel Urtasun, Antonio Torralba, and Sanja Fidler. Skip-thought vectors. In *Advances in neural information processing systems*, pp. 3294–3302, 2015.

Ashutosh Kumar, Satwik Bhattamishra, Manik Bhandari, and Partha Talukdar. Submodular optimization-based diverse paraphrasing and its effectiveness in data augmentation. In *NAACL-HLT*, 2019.

Wuwei Lan, Siyu Qiu, Hua He, and Wei Xu. A continuously growing dataset of sentential paraphrases. In *Proceedings of The 2017 Conference on Empirical Methods on Natural Language Processing (EMNLP)*, pp. 1235–1245. Association for Computational Linguistics, 2017. URL http://aclweb.org/anthology/D17-1127.

J Richard Landis and Gary G Koch. The measurement of observer agreement for categorical data. *biometrics*, pp. 159–174, 1977.

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

Chin-Yew Lin. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pp. 74–81, Barcelona, Spain, July 2004. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/W04-1013.

Dekang Lin and Patrick Pantel. Dirt@ sbt@ discovery of inference rules from text. In *Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 323–328, 2001.

Chang Liu, Daniel Dahlmeier, and Hwee Tou Ng. Pem: A paraphrase evaluation metric exploiting parallel texts. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, pp. 923–932, 2010.

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

Qingsong Ma, Ondřej Bojar, and Yvette Graham. Results of the wmt18 metrics shared task: Both characters and embeddings achieve good performance. In *Proceedings of the third conference on machine translation: shared task papers*, pp. 671–688, 2018.

Hongren Mao and Hungyi Lee. Polly want a cracker: Analyzing performance of parroting on paraphrase generation datasets. In *EMNLP/IJCNLP*, 2019.

Pedro Henrique Martins, Zita Marinho, and André F. T. Martins. Sparse text generation, 2020.

Ning Miao, Hao Zhou, Lili Mou, Rui Yan, and Lei Li. Cgmh: Constrained sentence generation by metropolis-hastings sampling. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pp. 6834–6842, 2019.

Tomas Mikolov, Kai Chen, Greg S. Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space, 2013. URL http://arxiv.org/abs/1301.3781.

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

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners, 2019. URL https://d4mucfpksywv.cloudfront.net/better-language-models/language_models_are_unsupervised_multitask_learners.pdf. Unpublished manuscript.

Aurko Roy and David Grangier. Unsupervised paraphrasing without translation. In *ACL*, 2019a.

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

Hubert JA Schouten. Nominal scale agreement among observers. *Psychometrika*, 51(3):453–466, 1986.

Thibault Sellam, Dipanjan Das, and Ankur P. Parikh. Bleurt: Learning robust metrics for text generation. In *ACL*, 2020.

Matthew G Snover, Nitin Madnani, Bonnie Dorr, and Richard Schwartz. Ter-plus: paraphrase, semantic, and alignment enhancements to translation edit rate. *Machine Translation*, 23(2-3): 117–127, 2009.

Hong Sun and Ming Zhou. 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)*, pp. 38–42, Jeju Island, Korea, July 2012. Association for Computational Linguistics. URL `https://www.aclweb.org/anthology/P12-2008`.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in neural information processing systems*, pp. 5998–6008, 2017.

Sean Welleck, Ilia Kulikov, Stephen Roller, Emily Dinan, Kyunghyun Cho, and Jason Weston. Neural text generation with unlikelihood training. *arXiv preprint arXiv:1908.04319*, 2019.

John Wieting and Kevin Gimpel. 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)*, pp. 451–462, 2018.

Wei Xu, Courtney Napoles, Ellie Pavlick, Quanze Chen, and Chris Callison-Burch. Optimizing statistical machine translation for text simplification. *Transactions of the Association for Computational Linguistics*, 4:401–415, 2016. doi: 10.1162/tacl_a_00107. URL `https://www.aclweb.org/anthology/Q16-1029`.

Rowan Zellers, Ari Holtzman, Hannah Rashkin, Yonatan Bisk, Ali Farhadi, Franziska Roesner, and Yejin Choi. Defending against neural fake news. In *Advances in Neural Information Processing Systems*, pp. 9051–9062, 2019.

Tianyi Zhang, V. Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. Bertscore: Evaluating text generation with bert. *ArXiv*, abs/1904.09675, 2020.

# A Appendix

## A.1 Derivation of Sampling Function

Here we derive the sampling function used for Reflective Decoding, working from a definition of similarity based on context and deriving a function that allows generation. This section is meant to supplement and expand upon §2.7.

To begin, for correctness we use the notation $P_{c|s}$ to denote the distribution of contexts $c$ for source sentence $s$. In practice, this will be 1-sided context, for instance right-hand context $c_{rh}$. In this case, $P_{c|s}$ would be estimated by the left-to-right language model conditioned on s: $\overrightarrow{\text{LM}}(c|s)$. A clarifying example is included in figure 1 where contexts are sampled from this distribution.

The reverse distribution $P_{s|c}$ is the opposite: going back from context *towards* text. With right-hand context, this is estimated by the the reverse language model $\overleftarrow{\text{LM}}(s|c)$. We will begin by using the more general notation ($P_{c|s}, P_{s|c}$) while considering theoretical quantities, then transition to using the language model distributions ($\overrightarrow{\text{LM}}$,$\overleftarrow{\text{LM}}$) when we are estimating these quantities in practice.

In §2.7, we consider the task of comparing a source sentence $s_{src}$ with another sentence $\hat{s}$. For instance, we may want to know if $\hat{s}$ is a paraphrase of $s_{src}$. Following a distributional intuition (Firth, 1957) that text with similar meaning will appear in similar contexts, we define a simple way to compare meaning

$$D_{KL}(P_{c|s_{src}}, P_{c|\hat{s}}) \tag{9}$$

Where $D_{KL}$ is the Kullback–Leibler measuring the difference between the distributions $P_{c|s_{src}}$ and $P_{c|\hat{s}}$. This is a simple way to capture a notion above: we take the amount the contexts of $s_{src}$ and $\hat{s}$ differ as a proxy for their difference in meaning and connotation.

While this equation (9) plays the same role as equation 5 in §2.7, we use the LM notation in that case, for simplicity and because we are only considering single-sided context there. Equation 9 is more general.

In a generation setting, we are most interested in selecting for contextual closeness, and therefore only need to rank between options. Therefore we will instead be working with the cross-entropy:

$$H(\overrightarrow{\mathrm{LM}}(c|s_{src}), \overrightarrow{\mathrm{LM}}(c|\hat{s})) = \sum_c -\overrightarrow{\mathrm{LM}}(c|s_{src}) log(P_{cj\hat{s}}(c)) \tag{10}$$

which is equivalent to $D_{KL}$ up to a constant offset, and will be easier to estimate in practice. Here, the sum over $c$ indicates a sum over every possible context $c$. In practice we will use a finite sample estimate, but will use the theoretical formulation for now.

As stated in Sec 2.7, we are using this as a measure of contextual difference in meaning. For the purposes of paraphrasing, we are trying to find a sentence $\hat{s}$ that minimizes this, which is equivalent to maximizing the exponent of its negation:

$$\begin{aligned} Score(\hat{s}) &= e^{\sum_c \quad P_{c|s} log(P_{c|\hat{s}}(c))} \\ &= \prod_c P_{cj\hat{s}}(c)^{P_{c|s}(c)} \\ &= \prod_c \left( \frac{P_{\hat{s}jc}(\hat{s})P(c)}{P(\hat{s})} \right)^{P_{c|s}(c)} \\ &= a_0 \prod_c \left( \frac{P_{\hat{s}jc}(\hat{s})}{P(\hat{s})} \right)^{P_{c|s}(c)} \\ &= \frac{a_0}{P(\hat{s})} \prod_c P_{\hat{s}jc}(\hat{s})^{P_{c|s}(c)} \end{aligned} \tag{11}$$

Note, $a_0$ is a constant factor resulting from factors of $P(c)$. We drop this for optimization. Also, $P_{\hat{s}jc}$ simply gives the distribution of text given context $c$ e.g. if $c$ is right context, this is the distribution estimated by a R2L LM conditioned on $c$.

The result of derivation 11 fits the Product of Experts model of Hinton (2002). In theory, the factor of $P(\hat{s})^{\ 1}$ will prioritize more context-specific paraphrases as a low probability sentence that's likely in contexts for $s$ is more related than a sentence that's generally likely (i.e. generic), but this has a few issues. For one, our estimators (language models) are not well equipped to handle very unlikely text, as they're trained on the real distribution and so spend relatively little capacity on very unlikely sequences. Second, while a less likely sentence can have higher similarity for the reasons stated above, this may not be the goal of our system.

In a real setting, we are interested in related sentences that are also *fluent* and *reasonable*. For this reason, we drop the $P(\hat{s})^{\ 1}$ term when calculating our approximate score, the equivalent of multiplying in $P(\hat{s})$ which is simply biasing the model towards likely sequences.

This gives an augmented score:

$$Score(\hat{s}) = c_0 \prod_c P_{\hat{s}jc}(\hat{s})^{P_{c|s}(c)} \tag{12}$$

Optimizing this is equivalent to taking a product of experts of the following form:

$$Score(\hat{s}) = \prod_c P_{\hat{s}jc}(\hat{s})^{w_{c|s}} \tag{13}$$

There is then the question of how we should set the weights $w_{cjs}$ in the limited sample setting. In the full setting, these weights are a set of probabilities $P_{cjs_{src}}(c)$ summing to 1 over all contexts c.

This indicates the logarithm of the score corresponds to a convex combination of the logits of the distributions $P_{\hat{s}jc}(\hat{s})$. To keep in line with this notion, we will also enforce that weights constitute a proper distribution.

In the limiting case with unlimited samples, these weights should be set to the probability of context given $s$, $P_{cjs}(c)$. A simple method to take this to the finite-sample case would be simply renormalizing these probabilities given over the sampled contexts. However, it is not clear that these are the most efficient weights for a good estimate of the scoring function. Further, while pretrained LMs are strong estimators, exponentiating by their estimates will magnify any errors they make. Instead, we learn these weights using a heuristic, discussed later.

As we transition to the finite-sample-setting and consider estimating this in practice, we replace the theoretical distributions with estimates using language models. In doing so, we go from a more general notion of context to 1-sided. Here we will consider right-context (meaning $P_{\hat{s}jc}$ is estimated by $\overleftarrow{\text{LM}}$) but the left-context case proceeds symmetrically. Substituting in the language model distribution:

$$Score(\hat{s}) = \prod_{c} \overleftarrow{\text{LM}}(\hat{s}|c)^{w_{c|s}} \tag{14}$$

Where now the product over $c$ indicates product over the finite sampled contexts. We will discuss how the weights are learned, but first we convert this to a sampling function. We can now decompose the scoring function into tokens of generation $\hat{s} = \hat{s}_0 ... \hat{s}_n$:

$$Score(\hat{s}_{0:n}) = \prod_{j} \prod_{c} \overleftarrow{\text{LM}}(\hat{s}_j|\hat{s}_{j+1:n})^{w_{c|\hat{s}}} \tag{15}$$

This is simply restating equation 13 but factorizing LM probability by tokens.

Renormalizing and decomposing by token position, this gives a natural distribution to sample from:

$$P_{sample}(\hat{s}_j|\hat{s}_{j+1:n}) = \frac{\prod_c \overleftarrow{\text{LM}}(\hat{s}_j|hats_{j+1:n})^{w_{c|s}}}{\sum_{t2V} \prod_c \overleftarrow{\text{LM}}(t|hats_{j+1:n})^{w_{c|s}}} \tag{16}$$

Simply, we are normalizing at each point over tokens in the vocabulary $V$, making this a proper token-wise distribution to sample from. Note that we are sampling right-to-left, so from index $n$ down, to match convention. This is the sampling function referred to as $\overleftarrow{\text{RD}}$ in the body of the paper, and state in equation 7. We use this to sample candidate generations that encourage adherence to the semantic scoring function. Note, in practice we refer to contexts by index $i$ ($c_i$) and the associated weight as $w_i$.

Finally, we learn the weights, following the convex combination constraint (weights are nonnegative, summing to 1), to match one fundamental aspect of the scoring function. That is, $s_{src}$ should receive the highest score (or similarly, should have the lowest contextual difference with itself). So essentially, we learn weights that maximize the score/probability of $s_{src}$, using a gradient-based learning algorithm to achieve this. This assures that the entire algorithm can proceed using only the two language models and input, as the signal for learning comes only from the self-generated context and input.

# B Implementation Details

## B.1 Left-to-Right Reflective Decoding sampling function

As mentioned in §2.3, a left-to-right Reflective Decoding sampling function $\overrightarrow{\text{RD}}$ is learned in a similar manner to $\overleftarrow{\text{RD}}$, simply by switching the roles of $\overrightarrow{\text{LM}}$ and $\overleftarrow{\text{LM}}$ in algorithm 1. For completeness we elaborate on this here.

First, the roles of the language models are flipped in the sampling function:

**Algorithm 2:** Learn REFLECTIVE DECODING sampling function (left-to-right)

**Input:** Left to right language model $\overrightarrow{\text{LM}}$
Right to left language model LM
Source text: $s_{src}$

1: Sample contexts, $c_1 ... c_{n_c}$ $\quad$ $\text{LM}(c|s_{src})$

2: Initialize parameters $\mathbf{w} = w_1 ... w_{n_c}$ s.t. $\sum w_i = 1, w_i \quad 0$

3: $\overrightarrow{\text{RD}}(s) \propto \prod_i \overrightarrow{\text{LM}}(s|c_i)^{w_i}$ normalized by token (equation 17)

4: learn $\mathbf{w} = \arg\max_{\mathbf{w}} \overrightarrow{\text{RD}}(s_{src})$
under $\sum w_i = 1, w_i \quad 0$

**Output:** $\overrightarrow{\text{RD}}$

$$\overrightarrow{\text{RD}}\,(\text{s}) = \frac{\prod_i \overrightarrow{\text{LM}}(s|c_i)^{w_i}}{\prod_{j=0}^{|s|} \sum_{t \in V} \prod_i \overrightarrow{\text{LM}}(t|s_{0:j \quad 1} + c_i)^{w_i}} \tag{17}$$

where contexts $c_i$ are now generated by the backwards language model $\overleftarrow{\text{LM}}$ (i.e. left-contexts). We then present algorithm 2, which defines how to learn $\overrightarrow{\text{RD}}$.

### B.2 POST-PROCESSING GENERATIONS

Without learning stop-tokens, REFLECTIVE DECODING samples fixed number ($len_{\hat{s}}$) of tokens. Candidates are extracted from raw generations using a combination of sentence tokenization to trim extra text to the sentence boundaries.

### B.3 ENTROPY CALIBRATION

Entropy calibration is used when sampling candidate generations (§2.4). We expand on the earlier definition here.

When sampling output generations, generation parameters (nucleus sampling $p_{\hat{s}}$ in paraphrasing) control how "greedy" or stochastic the sampling process is. However, the exact effect of a specific value of $p_{\hat{s}}$ depends on target length, number of generated contexts, complexity of meaning, desired level of agreement with source etc. Setting $p_{\hat{s}}$ too low may sample only the most likely option, but too high can result in off-topic candidates. Simply, the "correct" value of $p_{\hat{s}}$ is highly sample-dependent.

Instead, we define a technique, **entropy calibration**, designed to control how much "randomness" is used in sampling in a more robust way. Rather than directly setting a $p_{\hat{s}}$ for all examples, entropy calibration allows the user to specify the amount of randomness or approximate entropy $\hat{h}$ they would like to sample with over each example. In the greedy case for instance, the desired entropy $\hat{h}$ is set to 0, or equivalently we are picking from a set of 1 possible option. Likewise, the user might set $\hat{h}$ to 4, which would result in a higher $p_{\hat{s}}$ (how much higher will be example-dependant).

In practical terms, we search for $p_{\hat{s}}$ in each case that is expected to give the correct level of "randomness" over the entire generation, although $p_{\hat{s}}$ is a token-level parameter. To estimate how random a given value of $p_{\hat{s}}$ will make a generated sentence, we take the sampling entropy over the source string $s_0 ... s_n$ under the nucleus-sampling truncated distribution $\text{P}_p$:

$$\hat{h} = \sum_i \sum_{w \in V_p} -P_p(w|s_0 ... s_{i \quad 1}) \log P_p(w|s_0 ... s_{i \quad 1})$$

Where $V_p$ is the truncated vocabulary with parameter $p$. Roughly, this captures a branching-factor over the sequence. We select $p_{\hat{s}}$ that gives a desired entropy. We set this to values of 4 or 6 which we found effective. Parameters are available in App. B.4.

| Param | Paraphrasing | $\alpha$NLG |
|---|---|---|
| model size | Mega | Mega |
| $len_{\hat{s}}$ or $len_h$ | len$(s)+5$ | 20 |
| $len_c$ | 50 | 50 |
| $n_{\hat{s}}$ | 30 | 20 |
| $n_c$ | 80 | 50 |
| $h_{sample}$ | 4. | 6. |
| $p_c$ | 0.7 | 0.9 |
| $k_c$ | 6 | 6 |

Table 3: Most parameters are explained in §2.4. $h_{sample}$ is entropy for sampling calibration in §B.3

### B.4 PARAMETERS

In this section, we outline model settings for our 2 experimental settings, paraphrasing and $\alpha$NLG. See Table 3.

Broadly, $\alpha$NLG achieves higher variety in output with a higher sampling entropy ($h_{sample}$), more diverse generated contexts (higher $p_c$), and fewer generated contexts ($n_c$).

To find these parameters, we experimented with different reasonable values on the dev set of each model, and evaluated manually whether generated text appeared reasonable, specifically reading examples and picking model settings that produced good paraphrases as judged by the authors. For $len_{\hat{s}}$ and $len_h$ we simply set this high enough to ensure desirable outputs would be a subsequence.

We trained our transformer language models on TPU pods (using code in TensorFlow) of size 512 until convergence. During our generation experiments, we transferred the model to Pytorch, and ran locally on a machine with 2 NVIDIA Titan Xp GPUs.

## C EVALUATION

### C.1 AUTOMATIC METRICS

Links to the code we used for automatic evaluation is given here:

- ROUGE
- BLEU
- METEOR
- TER$_P$
- SARI
- BERTScore
- BLEURT

We include a limited number of metrics in the main paper for space and clarity, but include further metrics tested in table 4: ROUGE-1, ROUGE-2 (Lin, 2004), BLEURT Sellam et al. (2020), BERTScore (Zhang et al., 2020). For BLEURT, we used the ”BASE” pretrained model suggested by the authors. For BERTScore, we use the default settings of the official codebase.

### C.2 HUMAN EVALUATION

For human evaluation in paraphrasing, we evaluate on 204 input sentences, having 3 raters evaluate each model’s output. We ask about fluency, consistency with the source, and difference in wording from the source (template in figure 4). In each case, we use a multi-point likert scale, but are mainly interested in whether generations meet a threshold for each criterion: fluent enough to understand, with at most minor differences in meaning and at least minor differences in wording. In table 5 we give the rate for each model that humans found the generation meets these thresholds. The “Overall” column is the rate that humans find generations meet all 3 measured criteria. We take Fleiss’ $\kappa$ on

| | Method | R-1″ | R-2″ | BLEURT″ | BERTScore″ | $Novelty$ ″ |
|---|---|---|---|---|---|---|
| *Human* | Source | 70.1 | 47.0 | 19.9 | 95.2 | 0.0 |
| | Reference | 100.0 | 100.0 | 99.3 | 100.0 | 43.9 |
| *Supervised* | PG-IL | 66.6 | 44.0 | 11.1 | 94.7 | 24.4 |
| | DiPS | 56.7 | 33.7 | -29.5 | 92.7 | 48.5 |
| | BART | 63.6 | 41.6 | 9.6 | 94.4 | 35.2 |
| *Supervised (Bilingual)* | MT | 64.7 | 39.8 | 16.7 | 94.8 | 26.8 |
| *Unsupervised* | R-VQVAE | 68.2 | 32.0 | -7.6 | 93.2 | 26.2 |
| | $\text{CGMH}_{Top}$ | 55.6 | 29.6 | -53.6 | 92.1 | 27.6 |
| | $\text{CGMH}_{30}$ | 54.5 | 28.3 | -58.9 | 91.9 | 29.7 |
| | $\text{CGMH}_{45}$ | 48.5 | 22.1 | -80.9 | 90.7 | 44.5 |
| | $\text{RD}_{Top}$ (Us) | 65.8 | 42.3 | 15.3 | 94.8 | 20.8 |
| | $\text{RD}_{30}$ (Us) | 62.1 | 38.0 | 7.7 | 94.2 | 30.0 |
| | $\text{RD}_{45}$ (Us) | 56.8 | 31.1 | -1.9 | 93.5 | 45.0 |

Table 4: Model performance on the Quora test split. Included here are extra metrics beyond what is in the main paper. R-1 and R-2 refer to ROUGE-1 and ROUGE-2.

| | Human Quality″ | | | |
|---|---|---|---|---|
| Method | Fluency | Consistency | Novelty | Overall (%) |
| *Human* | | | | |
| Reference | 98.7 | 78.3 | 94.0 | 71.7 |
| *Supervised* | | | | |
| PG-IL | 95.9 | 79.9 | 51.0 | 29.4 |
| DiPS | 85.6 | 45.1 | 93.3 | 36.6 |
| BART | 97.2 | 77.6 | 68.8 | 46.1 |
| *Bilingual* | | | | |
| MT | 98.7 | 88.7 | 71.2 | 59.3 |
| *Unsupervised* | | | | |
| R-VQVAE | 84.2 | 76.3 | 60.3 | 33.5 |
| CGMH-Top | 79.4 | 43.1 | 85.6 | 27.0 |
| CGMH-30 | 78.8 | 37.9 | 96.4 | 31.5 |
| CGMH-45 | 71.6 | 19.9 | 98.5 | 15.8 |
| RD-Top (Us) | 98.0 | 84.6 | 43.5 | 27.5 |
| RD-30 (Us) | 98.7 | 75.3 | 88.2 | 63.2 |
| RD-45 (Us) | 97.5 | 67.3 | 95.3 | 62.1 |

Table 5: Model performance on the Quora test split, by human evaluation. Overall is calculated as the percentage of generations that meet the basic criteria of a paraphrase: fluent (the paraphrase can be understood), consistent with the source (the paraphrase shows at most **minor differences** in meaning from the source) and giving a novel phrasing (paraphrase shows at least **minor difference** in word choice). The first 3 columns indicate percentage of generations that meet the given criterion. Note, the first 3 rows (fluency, consistency, and novelty) are all required to for our notion of a good paraphrase, and each can be trivially maximized on its own.

these binary combined categories following Schouten (1986), finding agreement of 0.4 (fluency), 0.54 (consistency), 0.77 (wording), and 0.48 (overall) indicating moderate to substantial agreement (Landis & Koch, 1977).

As stated in §3, for $\alpha$NLG over 1000 examples we have 3 raters on Amazon Mechanical Turk evaluate each model. We ask about agreement between h and $o_1$, $o_2$, both, and overall quality on 4-value likert scales. We found Fleiss' kappa of 0.31, 0.29, 0.28, and 0.30 respectively, indicating fair agreement which is reasonable given the large number of options.

The template used for $\alpha$NLG is in figure 3, and for paraphrasing in figure 4.

| | Method | SARI" | BLEU" | METEOR" | TER$_P$# | $Novelty$ " |
|---|---|---|---|---|---|---|
| *Human* | Source | 13.6 | 36.7 | 25.0 | 75.0 | 0.0 |
| | Reference | 90.7 | 100.0 | 100.0 | 0.0 | 63.3 |
| *Supervised (Bilingual)* | MT | 36.1 | 29.4 | 22.1 | 80.0 | 30.4 |
| *Unsupervised* | R-VQVAE | 31.1 | 25.2 | 21.0 | 90.0 | 40.4 |
| | CGMH$_{Top}$ | 32.7 | 28.2 | 19.8 | 77.0 | 25.5 |
| | CGMH$_{30}$ | 33.2 | 26.3 | 18.7 | 78.0 | 30.1 |
| | CGMH$_{45}$ | 31.8 | 20.0 | 15.0 | 83.0 | 46.9 |
| | RD$_{Top/30}$ (Us) | 31.4 | 27.2 | 19.9 | 86.0 | 37.0 |
| | RD$_{45}$ (Us) | 36.4 | 25.5 | 18.9 | 89.0 | 47.0 |

Table 6: Model performance on the Twitter URL test split. **Bold** indicates best for model-type, * indicates best overall (excluding human). The first 4 columns are measures of quality, while the last measures novelty (equation 8) or dissimilarity from input. We rerun evaluations from past work. Note: Diversity of RD$_{Top}$ is over 30 and so this model is equivalent to RD$_{30}$ here.

### C.3 TWITTER DATASET

We include here a secondary paraphrasing evaluation on the Twitter URL corpus Lan et al. (2017), a set of paraphrase pairs created by linking tweets with matching shared URLs validated by human judges. This marks a significant domain shift from our primary paraphrasing task (question paraphrasing). We test the most comparable baselines to REFLECTIVE DECODING, mainly unsupervised models CGMH and R-VQVAE as well as the backtranslation MT model. These are all described in detail in §3.1. R-VQVAE the MT model, and REFLECTIVE DECODING do not use corpus-specific training data. CGMH trains on un-paired sentences from the Twitter training set, as outlined in the original work Miao et al. (2019). For all models, we use the same parameters as on the Quora dataset.

We include results on a number of automatic metrics in table 6. Results are similar to Quora: the most novel setting of Reflective Decoding (in this case RD$_{45}$) achieves the highest score on SARI, which seemed most aligned with Human on Quora and is the only metric included here that accounts for both novelty and quality. The metrics that do not account for novelty are unsurpisingly dominated by generations with lower novelty: $RD_{Top}$ gets the highest unsupervised BLEU (MT is highest overall), while R-VQVAE gets the highest unsupervised METEOR and CGMH$_{Top}$ the best unsupervised TER$_P$, while showing low levels of Novelty.

### C.4 REFLECTIVE SCORING EVALUATION

While the effectiveness of sampled contexts as an intermediate representation for generation is supported by our main experiments (§3), we would like to provide further validation of the semantic capacity of generated contexts, and validate the underlying scoring function of REFLECTIVE DECODING (equation 6).

Specifically, REFLECTIVE DECODING is based on the notion that generated contexts can capture the main semantic aspects of a source text. To explicitly test this, we measure how well the reflective scoring function of equation 6 captures semantic equivalence on the WMT18 metrics task (Ma et al., 2018). We specifically test on the segment-level evaluation, where the task is rate the semantic equivalence of a number of machine generated translations to a reference human translation. Metrics/scoring functions are evaluated by their rank correlation (using a metric related to Kendall's Tau Kendall (1938)) with human assessment of semantic equivalence. More details are available in the original task description.

In table 7, we present results on 3 language pairs: Chinese, German, and Estonian to English. Generally, higher correlation indicates a closer match with human judgement on which translations are closest in meaning to a reference translation. Metrics not significantly outperformed are bolded. We only include to-English translations as the language models used for REFLECTIVE DECODING are English.

To apply equation 6, we generate contexts for the reference, and test how well each generated translation fits the reference-contexts. Specifically, we follow equation 6, taking the reference

| Metric | cs ! en | de ! en | et ! en |
|---|---|---|---|
| Reflective Score | **0.357** | **0.490** | **0.364** |
| chrF+ | 0.288 | 0.479 | 0.332 |
| sentBLEU | 0.233 | 0.415 | 0.285 |
| CharacTER | 0.256 | 0.450 | 0.286 |
| BEER | 0.295 | 0.481 | 0.341 |
| ITER | 0.198 | 0.396 | 0.235 |
| RUSE | **0.347** | **0.498** | **0.368** |
| chrF | 0.288 | 0.479 | 0.328 |
| meteor++ | 0.270 | 0.457 | 0.329 |
| YiSi-1 | **0.319** | 0.488 | 0.351 |
| YiSi-0 | 0.301 | 0.474 | 0.330 |
| BLEND | **0.322** | 0.492 | 0.354 |
| YiSi-1$_{s}rl$ | **0.317** | **0.483** | 0.345 |
| UHH$_{T}SKM$ | 0.274 | 0.436 | 0.300 |

Table 7: Correlation with human judgement on the WMT18 metric task, for 3 language pairs (Chinese, German, and Estonian to English). Correlations of metrics not significantly outperformed by any other for that language pair are highlighted in bold. We note that Reflective Score is among only 2 methods bolded over all language-pairs tested.

sentence as $s_{src}$ and measure the contextual similarity with each translation (taking translation to be $\hat{s}$ in each case).

The high performance of our Reflective Scoring function (one of only 2 scoring functions bolded for all 3 pairs) indicates high agreement with human judges semantic equivalence between translations and reference. We are not aiming for state-of-the-art here, but rather to validate that generated contexts can represent full sentences well. This claim seems to be supported.

### C.5 Ablations

We include ablation studies for both the number of contexts generated $n_c$ (table 8) and whether weights $w$ are learned or set to be uniform (table 9). In both cases, besides ablated aspects the experiment is conducted as in §3.

For our ablation of generated contexts $n_c$ we investigate all 3 levels of novelty across 4 values of $n_c$: 6, 20, 40, 80, in table 8. We observe a broad trend of improving metrics with larger values of $n_c$, with the setting used in practice ($n_c = 80$) being a clear winner. An interesting aspect of this is that lower $n_c$ seems to force higher novelty. For $n_c = 20$, the lowest novelty achieved was 37.4, while this was 56.3 for $n_c = 6$. One potential cause for this is that the method cannot sufficiently capture the content of the source sentence with low $n_c$. With $n_c = 80$, Reflective Decoding can effectively rewrite the input if it's allowed to, but this doesn't seem to be true for lower $n_c$.

In ablating weight learning, we consider 2 possible values of $n_c$: 6 and 10. This is because, without weight learning we must set $k_c = n_c$ as we cannot do weight pruning if weights are uniform. For $k_c$ much higher than this, the ensemble of equation **??** becomes prohibitively expensive to calculate. For both values of $n_c$ we found RD with weight learning outperforms uniform weights on the SARI metric. Interestingly, we found uniform weights resulted in a *higher* novelty at $n_c = 6$ but a *lower* novelty at $n_c = 10$, with a gap of almost 50. With weight learning, this gap is only about 8, indicating less variability in behavior when weight learning is used.

## D Further Generations

See tables 11 - 15 for outputs of Reflective Decoding and a range of baselines on multiple example. We also include many generations of Reflective Decoding on the paraphrasing task in table 10

| Method | $n_c$ | SARI" | BLEU" | METEOR" | TER$_P$# | $Novelty$ " |
|---|---|---|---|---|---|---|
| RD$_{Top}$ | 80 | 29.0 | 49.9 | 33.9 | 52.0 | 20.8 |
| | 40 | 31.3 | 46.6 | 31.8 | 56.0 | 28.4 |
| | 20 | 34.1 | 43.0 | 29.8 | 61.0 | 37.4 |
| | 6 | 33.3 | 33.2 | 23.8 | 71.0 | 56.3 |
| RD$_{30}$ | 80 | 40.0 | 46.8 | 32.2 | 57.0 | 30.0 |
| | 40 | 39.4 | 44.2 | 30.6 | 59.0 | 35.6 |
| | 20 | 34.1 | 43.0 | 29.8 | 61.0 | 37.4 |
| | 6 | 33.3 | 33.2 | 23.8 | 71.0 | 56.3 |
| RD$_{45}$ | 80 | 38.6 | 39.9 | 28.9 | 65.0 | 45.0 |
| | 40 | 38.3 | 39.3 | 28.0 | 65.0 | 46.1 |
| | 20 | 38.5 | 39.6 | 28.0 | 65.0 | 45.8 |
| | 6 | 33.3 | 33.2 | 23.8 | 71.0 | 56.3 |

Table 8: Ablation of number of contexts generated $n_c$, holding weight pruning parameter constant at $k_c = 6$. For some $n_c$ (e.g. $n_c = 6$), the *Top* setting achieves high enough novelty for both cutoffs (30 and 45). In these cases, $RD_{Top}$ is repeated for $RD_{30}$ and $RD_{45}$

| Method | SARI" | BLEU" | METEOR" | TER$_P$# | $Novelty$ " |
|---|---|---|---|---|---|
| RD$_{Top}$ ($n_c = 6$ ) | 33.3 | 33.2 | 23.8 | 71.0 | 56.3 |
| - *weight learning* | 29.6 | 20.1 | 16.8 | 89.0 | 76.7 |
| RD$_{Top}$ ($n_c = 10$ ) | 34.2 | 37.3 | 26.4 | 65.0 | 48.2 |
| - *weight learning* | 31.3 | 46.6 | 31.8 | 56.0 | 28.4 |

Table 9: Ablation of whether weights are learned or taken to be uniform. For learned weights, we set $k_c = n_c$ indicating no weight pruning, so that the final number of contexts used is the same with and without learning.

| | |
|---|---|
| Input | what are the best books to expand imagination ? |
| RD$_{Top}$ | what are the best books to expand imagination ? |
| RD$_{30}$ | what are the best books to expand **our** imagination ? |
| RD$_{45}$ | what books **should you read** to expand **your** imagination ? |
| Input | how safe are ready to eat products ? |
| RD$_{Top}$ | how safe are ready to eat **foods and** products ? |
| RD$_{30}$ | how safe are ready to eat **foods and** products ? |
| RD$_{45}$ | how safe are foods that are ready to eat ? |
| Input | what can be the future of kashmir ? |
| RD$_{Top}$ | what **is** the future of kashmir ? |
| RD$_{30}$ | what **is** the future of kashmir ? |
| RD$_{45}$ | what **will** be the **fate** of kashmir ? |
| Input | does eating eggs also cause bird flu ? |
| RD$_{Top}$ | **is it true that** eating eggs **can cause** bird flu ? |
| RD$_{30}$ | **is it true that** eating eggs **can cause** bird flu ? |
| RD$_{45}$ | **is it true that** eating eggs **can cause** bird flu ? |
| Input | how do you open odt files on word ? |
| RD$_{Top}$ | how do you open odt files on word ? |
| RD$_{30}$ | how do you open odt files on **microsoft** word ? |
| RD$_{45}$ | how do you open an odt file on word ? |
| Input | if someone rejects your facebook friend request can you send it again ? |
| RD$_{Top}$ | if someone rejects your facebook friend request can you send it again ? |
| RD$_{30}$ | if someone rejects your facebook friend request can you message them again ? |
| RD$_{45}$ | what if someone accepts your facebook friend request ? |
| Input | when will science prove that god exists ? |
| RD$_{Top}$ | when will science prove that god exists ? |
| RD$_{30}$ | when will science prove **the existence of** god ? |
| RD$_{45}$ | when will science prove **the existence of** god ? |
| Input | is it good to use hair dryers ? |
| RD$_{Top}$ | is it good to use hair dryers ? |
| RD$_{30}$ | is it good to use **a** hair dryer ? |
| RD$_{45}$ | do you know that it 's good to use hair dryers ? |
| Input | is demonetizing notes of 500 and 1000 a permanent solution to curb black money ? |
| RD$_{Top}$ | is it a permanent solution to curb black money ? |
| RD$_{30}$ | is it a permanent solution to curb black money ? |
| RD$_{45}$ | **do you think that** demonetizing rs 500 and rs 1000 notes **is going to solve the problem of** black money ? |
| Input | what 's the best way to ask out a girl at my school ? |
| RD$_{Top}$ | what 's the best way to ask out a girl **in** my school ? |
| RD$_{30}$ | what 's the best way to ask out a girl **in** my school ? |
| RD$_{45}$ | how do i ask a girl out at school ? |
| Input | what was meant by the final scene in 2001 : a space odyssey ? |
| RD$_{Top}$ | what was meant by the final scene in 2001 : a space odyssey ? |
| RD$_{30}$ | what did he mean by the final scene in 2001 : a space odyssey ? |
| RD$_{45}$ | what did **stanley kubrick** mean by the final scene in **2001** ? |
| Input | how can i help my dog get rid of hiccups ? |
| RD$_{Top}$ | how can i help my dog get rid of hiccups ? |
| RD$_{30}$ | how can i help my dog get rid of **the** hiccups ? |
| RD$_{45}$ | can i help my dog **with hiccups** ? |

Table 10: Generations from the 3 REFLECTIVE DECODING variants on the examples from the Quora test split. Places where a novel phrasing is introduced are **bolded** and mistakes are underlined.

| Sam didn't sleep well last night. ___ Sam felt embarrassed and was planning on going to bed early that day. | |
|---|---|
| **Human** | Sam treated people poorly due to lack of sleep. |
| Unsupervised | |
| **REFLECTIVE DECODING** | He was so tired that he couldn't even get out of bed. |
| **GPT2-fixed** | he was really tired from finishing his book |
| Supervised | |
| **COMeT-Emb+GPT2** | Sam was still tired when she woke up. |
| **COMeT-Txt+GPT2** | Sam was too tired to finish his homework. |
| $O_1$-$O_2$-**Only** | Sam was very tired after dinner. |

Table 11: Further generations for $\alpha$NLG from all tested systems

| Cindy got out of bed. ___ She also ordered an iPhone. | |
|---|---|
| **Human** | She ordered a new iPad. |
| Unsupervised | |
| **REFLECTIVE DECODING** | On her way out, she ordered an iPad, one that was small enough to fit in her apartment. |
| **GPT2-fixed** | I'm sure, given all the likely possibilities of electrical shocks and heart attacks it's a good thing |
| Supervised | |
| **COMeT-Emb+GPT2** | Cindy went on Facebook. |
| **COMeT-Txt+GPT2** | Cindy liked the new slim iPhone. |
| $O_1$-$O_2$-**Only** | Cindy saw she didn't have enough money for clothes. |

Table 12: Further generations for $\alpha$NLG from all tested systems

| I once knew a girl named Sammy. ___ She got help and everything was alright. | |
|---|---|
| **Human** | She suffered from depression. |
| Unsupervised | |
| **REFLECTIVE DECODING** | She had problems and she needed help. |
| **GPT2-fixed** | I didn't think to her, this was a normal situation |
| Supervised | |
| **COMeT-Emb+GPT2** | Sammy was in a car accident. |
| **COMeT-Txt+GPT2** | Sammy got into a bad accident and her car broke down. |
| $O_1$-$O_2$-**Only** | Sammy got hit by a drunk driver. |

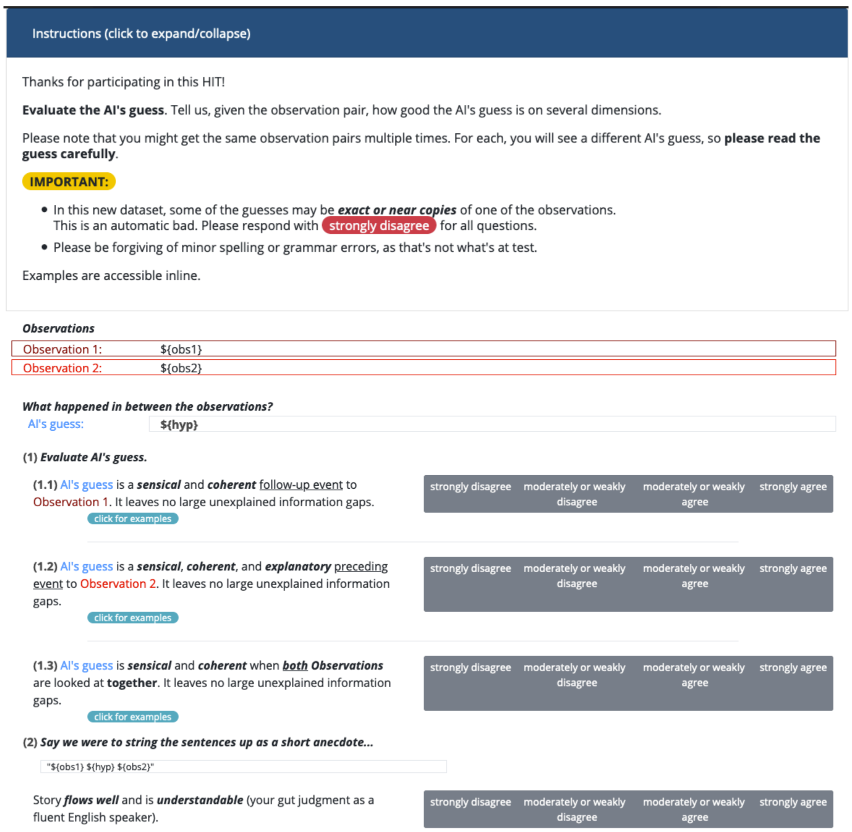
Table 13: Further generations for $\alpha$NLG from all tested systems

| Can you trust the information on Quora? | |
|---|---|
| **Human** | Do you trust Quora? |
| Unsupervised | |
| **RefDec-Top (Us)** | Can you trust the information on Quora? |
| **RefDec-70 (Us)** | Can I trust the information on Quora? |
| **RefDec-55 (Us)** | When can I trust information on Quora ? |
| **R-VQVAE** | Can you trust the information on Quora? |
| **CGMH-Top** | Can you answer the information on Quora? |
| **CGMH-70** | Can you answer the information on Quora? |
| **CGMH-55** | Can you answer more topics on Quora? |
| Supervised | |
| **PG-IL** | Can you trust the information on Quora? |
| **DiPS** | Can we trust our questions in Quora? |
| **BART** | Can you trust everything you read on Quora? |
| Bilingual | |
| **MT** | Can you trust the information on Quora? |

Table 14: Further generations for paraphrasing from all tested systems

| What is your creative process? | |
|---|---|
| **Human** | What's your creative process? |
| Unsupervised | |
| **RefDec-Top (Us)** | What is your creative process? |
| **RefDec-70 (Us)** | What's your creative process? |
| **RefDec-55 (Us)** | What's your creative process like? |
| **R-VQVAE** | What is your creative process? |
| **CGMH-Top** | What is your dream key? |
| **CGMH-70** | What is your dream key? |
| **CGMH-55** | What is your dream key? |
| Supervised | |
| **PG-IL** | What is your creative process? |
| **DiPS** | What is your creative strategy? |
| **BART** | What is your creative process? |
| Bilingual | |
| **MT** | What is your creative process? |

Table 15: Further generations for paraphrasing from all tested systems



Instructions (click to expand/collapse)

Thanks for participating in this HIT!

**Evaluate the AI's guess**. Tell us, given the observation pair, how good the AI's guess is on several dimensions.

Please note that you might get the same observation pairs multiple times. For each, you will see a different AI's guess, so **please read the guess carefully**.

**IMPORTANT:**

- In this new dataset, some of the guesses may be ***exact or near copies*** of one of the observations. This is an automatic bad. Please respond with strongly disagree for all questions.
- Please be forgiving of minor spelling or grammar errors, as that's not what's at test.

Examples are accessible inline.

***Observations***

| | |
|---|---|
| Observation 1: | ${obs1} |
| Observation 2: | ${obs2} |

***What happened in between the observations?***

| | |
|---|---|
| AI's guess: | **${hyp}** |

**(1)** ***Evaluate AI's guess.***

**(1.1)** AI's guess is a ***sensical*** and ***coherent*** follow-up event to Observation 1. It leaves no large unexplained information gaps.
click for examples

strongly disagree | moderately or weakly disagree | moderately or weakly agree | strongly agree

**(1.2)** AI's guess is a ***sensical***, ***coherent***, and ***explanatory*** preceding event to Observation 2. It leaves no large unexplained information gaps.
click for examples

strongly disagree | moderately or weakly disagree | moderately or weakly agree | strongly agree

**(1.3)** AI's guess is ***sensical*** and ***coherent*** when ***both Observations*** are looked at **together**. It leaves no large unexplained information gaps.
click for examples

strongly disagree | moderately or weakly disagree | moderately or weakly agree | strongly agree

**(2)** ***Say we were to string the sentences up as a short anecdote...***

"${obs1} ${hyp} ${obs2}"

Story ***flows well*** and is ***understandable*** (your gut judgment as a fluent English speaker).

strongly disagree | moderately or weakly disagree | moderately or weakly agree | strongly agree

Figure 3: The template used for human evaluation of $\alpha$NLG

**Instructions (click to expand/collapse)**

Thanks for participating in this HIT! Please read the instructions carefully.

In this HIT, you will be shown a **source sentence** and a **paraphrase** of that text.
Your task is to ***evaluate*** the **paraphrase** along three dimensions:

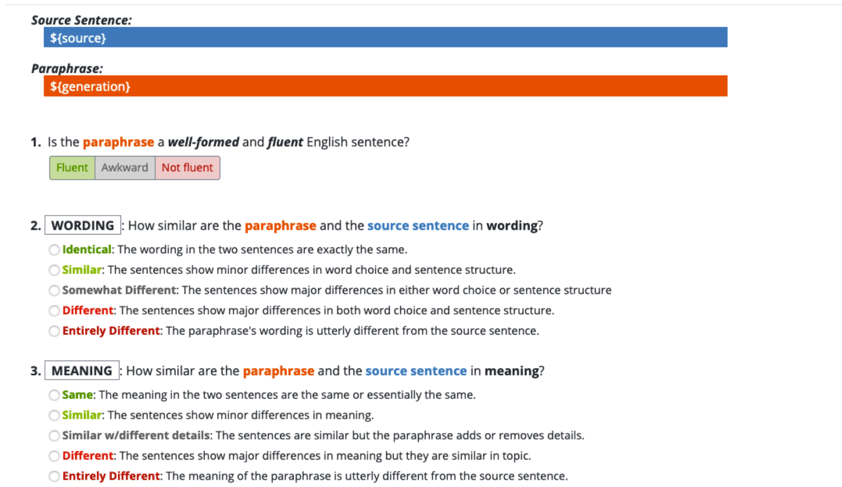
1. Is the **paraphrase** a ***well-formed*** and ***fluent*** English sentence?
   - **Yes**: The paraphrase is well-formed and fluent.
   - **Neither**: I understand the paraphrase, but is a bit *awkward* in its phrasing.
   - **No**: The paraphrase is neither well-formed or fluent.
2. Do the **paraphrase** and the **source sentence** have the ***same wording*** ?
   - **Identical**: The wording in the two sentences are **exactly the same**.
     *ex: "Do I buy a laptop first?" is **identical** to "Do I buy a laptop first ?"*
   - **Similar**: The two sentences show **minor differences** in word choice and sentence structure.
     *ex: "Do I buy a laptop first?" is **similar** to "Do I buy a computer first?"*
   - **Somewhat Different**: The two sentences show **major differences** in **either** word choice or sentence structure.
     *ex: "Do I buy a laptop first?" is **somewhat different** from "Do I buy an icecream first?" (word choice is very different, but the structure is the same)*
     *"Do I buy a laptop first?" is **somewhat different** from "I should first buy a laptop" (structure is quite different, but words are the similar)*
   - **Different**: The two sentences show **major differences** in **both** word choice and sentence structure.
     *ex: "Do I buy a laptop first?" is **different** from "Should I first purchase a laptop computer?"*
     *"Do I buy a laptop first?" is **different** from "I should first buy several ladders"*
   - **Entirely Different**: The paraphrase's wording **utterly different** from the source sentence.
     *ex: "Do I buy a laptop first?" is **entirely different** from "The first problem is the conclusion"*
3. Do the **paraphrase** and the **source sentence** have the ***same meaning*** ?
   - **Same**: The meaning in the two sentences are same or essentially the same.
     ⇒ if two sentences have **identical wording** then they also have **same meaning**. New
   - **Similar**: The sentences show minor differences in meaning.
   - **Similar w/different details**: The sentences are similar but the paraphrase adds or removes details. New
   - **Different**: The sentences show major differences in meaning but they are similar in topic.
   - **Entirely Different**: The meaning in the paraphrase is utterly different from the source sentence.

Please take care to not submit responses that are uninformed by the instructions.

**Examples for Question 2: Similarity assessment (click to expand/collapse)**

| Source sentence | Paraphrase | Wording | Meaning |
|---|---|---|---|
| A bird is bathing in the bird bath | A bird is bathing in the bird bath | **Identical**: sentences are exactly the same. | **Same**: since they are worded the same, they must mean the same thing! |
| | A winged creature is bathing in a water bath | **Similar**: There is a difference in word choice, but the difference is minor. | **Similar**: Meaning becomes more general (say a bat is an winged creature too), but still they are fairly similar. |
| | Birds bathe once a week | **Different** | **Different**: Meaning is different, but share in topic (about birds bathing) |
| | Cows are chewing cud | **Entirely Different** | **Entirely Different**: How is this even a paraphrase? |
| how do I know what he feels about me? | how do I know what he feels about me ? | **Identical** | **Same** |
| | how do I feel about him? | **Similar** | **Different** |
| | how do I know what he thinks about me? | **Similar** | **Same** |
| | How do I know if he really hates me? | **Similar** | **Similar** |

Figure 4: The template used for human evaluation of paraphrasing (part 1 of 2)



***Source Sentence:***

${source}

***Paraphrase:***

${generation}

1. Is the **paraphrase** a ***well-formed*** and ***fluent*** English sentence?

   Fluent | Awkward | Not fluent

2. WORDING : How similar are the **paraphrase** and the **source sentence** in **wording**?
   - **Identical**: The wording in the two sentences are exactly the same.
   - **Similar**: The sentences show minor differences in word choice and sentence structure.
   - **Somewhat Different**: The sentences show major differences in either word choice or sentence structure
   - **Different**: The sentences show major differences in both word choice and sentence structure.
   - **Entirely Different**: The paraphrase's wording is utterly different from the source sentence.

3. MEANING : How similar are the **paraphrase** and the **source sentence** in **meaning**?
   - **Same**: The meaning in the two sentences are the same or essentially the same.
   - **Similar**: The sentences show minor differences in meaning.
   - **Similar w/different details**: The sentences are similar but the paraphrase adds or removes details.
   - **Different**: The sentences show major differences in meaning but they are similar in topic.
   - **Entirely Different**: The meaning of the paraphrase is utterly different from the source sentence.

Figure 5: The template used for human evaluation of paraphrasing (part 2 of 2)