NeuroLogic A★esque Decoding: Constrained Text Generation with Lookahead Heuristics

Ximing Lu[‡†] ♡**Sean Welleck**[†‡] ♡**Peter West**[†]
Liwei Jiang[‡†] **Jungo Kasai**[‡†] **Daniel Khashabi**[‡] **Ronan Le Bras**[‡]
Lianhui Qin[†] **Youngjae Yu**[‡] **Rowan Zellers**[†] **Noah A. Smith**[†‡] **Yejin Choi** [†‡]
[‡]Allen Institute for Artificial Intelligence
[†]Paul G. Allen School of Computer Science & Engineering, University of Washington

Abstract

The dominant paradigm for neural text generation is left-to-right decoding from autoregressive language models. Constrained or controllable generation under complex lexical constraints, however, requires foresight to plan ahead feasible future paths.

Drawing inspiration from the A* search algorithm, we propose NeuroLogic A★esque,[1] a decoding algorithm that incorporates heuristic estimates of future cost. We develop efficient *lookahead* heuristics that are efficient for large-scale language models, making our method a drop-in replacement for common techniques such as beam search and top-k sampling. To enable constrained generation, we build on NeuroLogic decoding (Lu et al., 2021), combining its flexibility in incorporating logical constraints with A★esque estimates of future constraint satisfaction.

Our approach outperforms competitive baselines on five generation tasks, and achieves new state-of-the-art performance on table-to-text generation, constrained machine translation, and keyword-constrained generation. The improvements are particularly notable on tasks that require complex constraint satisfaction or in few-shot or zero-shot settings. NeuroLogic A★esque illustrates the power of decoding for improving and enabling new capabilities of large-scale language models.

1 Introduction

The dominant paradigm for neural text generation is based on left-to-right decoding from autoregressive language models such as GPT-2/3 (Radford et al., 2019; Brown et al., 2020). Under this paradigm, common decoding techniques such as beam search or top-k/p sampling (Holtzman et al., 2020) determine which token to generate

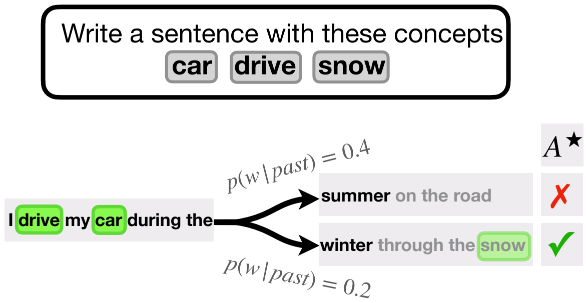


Figure 1: NeuroLogic★ leverages *lookahead heuristics* to guide generations towards those that satisfy the given task-specific constraints. In this example from the CommonGen task, although **summer** is a *more likely* next word given the already-generated *past*, NeuroLogic★ looks ahead to see that selecting **winter** results in a generation that incorporates unsatisfied constraint **snow** with a higher probability later on. Thus, **winter** is preferred despite being lower probability than **summer**.

next based on what happened in the past, without explicitly looking ahead into the future. While this lack of foresight often suffices for open-ended text generation – where any coherent text can be acceptable – for constrained text generation, planning ahead is crucial for incorporating all desired content in the generated output (Hu et al., 2017; Dathathri et al., 2019).

Classical search algorithms such as A* search (Hart et al., 1968; Pearl, 1984; Korf, 1985) address the challenge of planning ahead by using *heuristic* estimation of future cost when making decisions. Drawing inspiration from A* search, we develop NeuroLogic A★esque (shortened to NeuroLogic★), which combines A*-like heuristic estimates of future cost (e.g. perplexity, constraint satisfaction) with common decoding algorithms for neural text generation (e.g. beam search, top-k sampling), while preserving the efficiency demanded by large-scale neural language models.

As selecting the next token to generate based on the *optimal* future cost is NP-complete (Chen et al.,

♡ Co-second-authors. Other authors are listed alphabetically, as all contributed significantly.

[1]pronounced [ey stɑr ɛsk].

2018), we develop *lookahead* heuristics, which approximate cost at each decoding step based on continuations of the sequence-so-far. Figure 1 shows an example, where NEUROLOGIC A★esque guides generation towards a decision that would have been ignored based on the past alone, but is selected after looking ahead and incorporating the probability that constraints are satisfied in the future.

Our approach builds on NEUROLOGIC Decoding of Lu et al. (2021), a variation of beam-search for controlling generation through rich logic-based lexical constraints expressed in Conjunctive Normal Form (CNF). Our work generalizes Lu et al. (2021) by (1) incorporating novel lookahead heuristics to estimate future contraint satisfaction, and (2) developing additional *unconstrained* variants that can work with an empty set of constraints. These new algorithm variants support broad applications of NEUROLOGIC★, including unconstrained generation, as demonstrated in our experiments.

Extensive experiments across five generation tasks demonstrate that our approach outperforms competitive baselines. We test NEUROLOGIC★ in conjunction with both supervised and unsupervised models and find that the performance gain is pronounced especially in zero-shot or few-shot settings. In particular, on the COMMONGEN benchmark, using our proposed decoding algorithm with an off-the-shelf language model outperforms a host of *supervised* baselines with conventional decoding algorithms. This demonstrates that a strong inference-time algorithm such as NEUROLOGIC★ can alleviate the need for costly datasets that are manually annotated for explicit supervision. Moreover, we find that NEUROLOGIC★ achieves state-of-the-art performance in various settings, including WMT17 English-German machine translation with lexical constraints (Dinu et al., 2019) and few-shot E2ENLG table-to-text generation (Chen et al., 2020b).

In summary, we develop NEUROLOGIC A★esque, a new decoding algorithm for effective and efficient text generation. To our knowledge this is the first A*-like algorithm for guided text generation via lookahead heuristics. Our algorithm is versatile, as it can be applied to a variety of tasks via inference-time constraints, reducing the need for costly labeled data. Extensive experiments show its effectiveness on several important generation benchmarks.

2 NEUROLOGIC A★esque Decoding

We describe NEUROLOGIC A★esque Decoding (shortened as NEUROLOGIC★), our decoding algorithm motivated by A* search (Hart et al., 1968), a best-first search algorithm that finds high-scoring paths using a heuristic estimate of future return. We first introduce the decoding problem, and then describe our heuristics with a novel lookahead procedure for adapting NEUROLOGIC★ search to unconstrained and constrained generation with large-scale autoregressive models.

2.1 Decoding With A★esque Lookahead

Decoding. Sequence-to-sequence generation is the task of generating an output sequence $\mathbf{y}$ given an input sequence $\mathbf{x}$. We consider standard left-to-right, autoregressive models, $p_\theta(\mathbf{y}|\mathbf{x}) = \prod_{t=1}^{|\mathbf{y}|} p_\theta(y_t|\mathbf{y}_{<t}, \mathbf{x})$, and omit $\mathbf{x}$ to reduce clutter. Decoding consists of solving,

$$\mathbf{y}_* = \arg\max_{\mathbf{y}\in\mathcal{Y}} F(\mathbf{y}). \tag{1}$$

Where $\mathcal{Y}$ is the set of all sequences. In our setting, the objective $F(\mathbf{y})$ takes the form $s(\mathbf{y}) + H(\mathbf{y})$, where $s(\mathbf{y})$ is $\log p_\theta(\mathbf{y})$, and $H(\mathbf{y})$ is either zero or is a score for satisfying constraints on $\mathbf{y}$.

Our method takes the perspective of decoding as discrete search, in which states are partial prefixes, $\mathbf{y}_{<t}$, actions are tokens in vocabulary $\mathcal{V}$ (i.e. $y_t \in \mathcal{V}$) and transitions add a token to a prefix, $\mathbf{y}_{<t} \circ y_t$. Each step of decoding consists of 1) expanding a set of candidate next-states, 2) scoring each candidate, and 3) selecting the k best candidates:

$$\begin{aligned} Y'_t &= \{\mathbf{y}_{<t} \circ y_t \mid \mathbf{y}_{<t} \in Y_{t-1}, y_t \in \mathcal{V}\}, \\ Y_t &= \underset{(\mathbf{y}_{<t}, y_t)\in Y'_t}{\arg\operatorname{topk}} \{f(\mathbf{y}_{<t}, y_t)\}, \end{aligned} \tag{2}$$

where $Y_0 = \{\langle bos \rangle\}$ and $f(\cdot)$ is a scoring function that approximates the objective F. Common decoding algorithms such as beam search score candidates without considering future tokens, e.g., $f(\mathbf{y}_{<t}, y_t) = \log p_\theta(\mathbf{y}_{\leq t})$.

Lookahead heuristics. Our method incorporates an estimate of the future into candidate selection. Ideally, we want to select candidates that are on optimal trajectories, replacing Equation 2 with:

$$Y_t = \underset{(\mathbf{y}_{<t}, y_t)\in Y'_t}{\arg\operatorname{topk}} \left\{ \max_{\mathbf{y}_{>t}} F(\mathbf{y}_{<t}, y_t, \mathbf{y}_{>t}) \right\}. \tag{3}$$

However, computing Equation 3 presents two difficulties: 1) the objective $F(\mathbf{y})$ may be unknown or difficult to compute, and 2) the space of future trajectories $\mathbf{y}_{>t}$ is prohibitively large.

Motivated by A* search (Hart et al., 1968), a best-first search algorithm that finds high-scoring paths by selecting actions that maximize,

$$f(a) = s(a) + h(a),$$

where $s(a)$ is the score-so-far and $h(a)$ is a heuristic estimate of the future score. We approximate the objective using a lightweight *heuristic* $h(\cdot)$,

$$Y_t = \underset{\mathbf{y}_{\leq t} \in Y'_t}{\arg \operatorname{topk}} \left\{ s(\mathbf{y}_{\leq t}) + \max_{\mathbf{y}_{>t}} h(\mathbf{y}_{<t}, y_t, \mathbf{y}_{>t}) \right\}, \tag{4}$$

where $s(\mathbf{y}_{\leq t}) = \log p_\theta(\mathbf{y}_{\leq t})$. To make the search tractable, we search over a set of *lookahead* continuations, approximating Equation 3 as,

$$Y_t = \underset{\mathbf{y}_{\leq t} \in Y'_t}{\arg \operatorname{topk}} \left\{ s(\mathbf{y}_{\leq t}) + \max_{\mathcal{L}_\ell(\mathbf{y}_{\leq t})} h(\mathbf{y}_{\leq t+\ell}) \right\}, \tag{5}$$

where each element $\mathbf{y}_{t+1:t+\ell}$ of $\mathcal{L}_\ell(\mathbf{y}_{\leq t})$ is a length-ℓ continuation of $\mathbf{y}_{\leq t}$. Beam search corresponds to setting ℓ and h to 0.

A*esque decoding. Beam search, A* search, and our method fall under a general class of algorithms that differ based on (1) which candidates are expanded, (2) which candidates are pruned, (3) how candidates are scored (Meister et al., 2020). We inherit the practical advantages of beam search-style expansion and pruning, while drawing on A*-like heuristics to incorporate estimates of the future, and refer to our method as **A*esque decoding**.

Generating lookaheads. We compare several methods for generating the lookaheads $\mathcal{L}_\ell(\mathbf{y}_{\leq t})$.

The *greedy* lookahead produces a single sequence, $\mathcal{L}_\ell = \{\mathbf{y}_{t+1:t+\ell}\}$, starting from $\mathbf{y}_{\leq t}$ and selecting each token according to $y_{t'} = \arg\max_{y \in \mathcal{V}} p_\theta(y|\mathbf{y}_{<t'})$.

We also consider a relaxation which interpolates between providing the greedy token and a uniform mixture of tokens as input at each step. Specifically, we adjust the model's probabilities with a temperature, $\tilde{p}_\theta(y_t|\mathbf{y}_{<t}) = \operatorname{softmax}(s_t/\tau)$, where $s_t \in \mathbb{R}^{|\mathcal{V}|}$ is a vector of logits, and feed the expected token embedding as input at step t,

$$e_t = \mathbb{E}_{y_t \sim \tilde{p}(y_t|\mathbf{y}_{<t})}[E(y_t)], \tag{6}$$

where $E \in \mathbb{R}^{|\mathcal{V}| \times d}$ is the model's token embedding matrix. This *soft lookahead* moves from providing the greedy token as input ($\tau \to 0$) to a uniform mixture of tokens ($\tau \to \infty$) based on the value of temperature τ. When using the soft lookahead, we use $\tilde{p}$ in place of p when scoring tokens. The soft (and greedy) lookahead is efficient, but only explores a single trajectory.

The *beam* lookahead trades off efficiency for exploration, returning a set $\mathcal{L}_\ell$ containing the top-k candidates obtained by running beam search for ℓ steps starting from $\mathbf{y}_{<t}$.

Finally, the *sampling* lookahead explores beyond the highly-probable beam search continuations, generating each $\mathbf{y}_{t+1:t+\ell} \in \mathcal{L}_\ell$ using,

$$y_{t'} \sim p_\theta(y|\mathbf{y}_{<t'}),$$

for t' from t+1 to t+k.

Next, we move to our proposed lookahead heuristics, starting with the unconstrained setting.

2.2 Unconstrained Generation with NeuroLogic*

First we consider a standard decoding setting,

$$\arg\max_{\mathbf{y} \in \mathcal{Y}} \log p_\theta(\mathbf{y}|\mathbf{x}).$$

We score candidates based on a combination of the *history* and *estimated future*, by using the likelihood of the lookahead as a heuristic. That is, at the tth step of decoding, we use Equation 5:

$$h(\mathbf{y}_{\leq t+\ell}) = \lambda \log p_\theta(\mathbf{y}_{t+1:t+\ell}|\mathbf{y}_{\leq t}, \mathbf{x}), \tag{7}$$

where λ controls how much we rely on the estimated future versus the history, similar to weighted A* (Pohl, 1970).

2.3 NeuroLogic* for Constrained Generation

Our lookahead heuristics lend themselves to decoding with lexical constraints in a way that standard beam search does not. For constrained generation, we build on and generalize NeuroLogic decoding algorithm of Lu et al. (2021)– a beam-based search algorithm that supports a wide class of logical constraints for lexically constrained generation– with estimates of future contraint satisfaction.

Background of NeuroLogic. NeuroLogic Lu et al. (2021) accepts lexical constraints in Conjunctive Normal Form (CNF):

$$\underbrace{(D_1 \vee D_2 \cdots \vee D_i)}_{C_1} \wedge \cdots \wedge \underbrace{(D_{i'} \vee D_{i'+1} \cdots \vee D_N)}_{C_M}$$

where each D_i represents a single positive or negative constraint, $D(\mathbf{a}, \mathbf{y})$ or $\neg D(\mathbf{a}, \mathbf{y})$, enforcing the phrase **a** to be included in or omitted from **y**. Lu et al. (2021) refer to each constraint D_i as a *literal*, and each disjunction C_j of literals as a *clause*.

NEUROLOGIC is a beam-based approximate search for an objective which seeks fluent sequences in which all clauses are satisfied:

$$\arg\max_{\mathbf{y}\in\mathcal{Y}} p_\theta(\mathbf{y}|\mathbf{x}) - \lambda' \sum_{j=1}^{M}(1 - C_j),$$

where $\lambda' \gg 0$ penalizes unsatisfied clauses. At each step of the search, NEUROLOGIC scores each of the $k \times |\mathcal{V}|$ candidates $(\mathbf{y}_{<t}, y_t)$ based on whether they (partially) satisfy new constraints,

$$f(\mathbf{y}_{\leq t}) = \log p_\theta(\mathbf{y}_{\leq t}|\mathbf{x}) + \lambda_1 \max_{D(\mathbf{a},\mathbf{y}_{\leq t})} \frac{|\hat{\mathbf{a}}|}{|\mathbf{a}|}, \quad (8)$$

where the maximization is over a set of unsatisfied multi-token constraints **a** tracked by NEUROLOGIC, and $\hat{\mathbf{a}}$ is the prefix of **a** in the ongoing generation. For example, for $\mathbf{y}_{\leq t}$ =“*The boy climbs an apple*” and constraint **a**=“*apple tree*”, $\hat{\mathbf{a}}$ is “*apple*”. Intuitively, this function rewards candidates that are in the process of satisfying a constraint.

In lieu of taking the top-k scoring candidates (Equation 5), NEUROLOGIC prunes candidates that contain clauses that violate constraints, groups the candidates to promote diversity, and selects high-scoring candidates from each group. We use the same pruning and grouping approach, and refer the reader to Lu et al. (2021) for further details.

NEUROLOGIC★ decoding. Our method improves upon the NEUROLOGIC scoring function with an estimate of future constraint satisfaction. Our key addition is a lookahead heuristic that adjusts a candidate $(\mathbf{y}_{<t}, y_t)$'s score proportional to the probability of satisfying additional constraints in the lookahead $\mathbf{y}_{t+1:t+\ell}$:

$$h_{\text{future}}(\mathbf{y}_{\leq t+\ell}) = \lambda_2 \max_{D(\mathbf{a},\mathbf{y}_{\leq t})} \log p_\theta(D(\mathbf{a}, \mathbf{y}_{t+1:t+\ell})|\mathbf{x}, \mathbf{y}_{\leq t}), \quad (9)$$

where we define the probability that constraint **a** is satisfied using the most probable subsequence,

$$p_\theta(D(\mathbf{a}, \mathbf{y}_{t+1:t+\ell})|\mathbf{x}, \mathbf{y}_{\leq t}) = \max_{t'\in[t,t+\ell]} p_\theta(\mathbf{y}_{t':t'+|\mathbf{a}|} = \mathbf{a}|\mathbf{x}, \mathbf{y}_{<t'}), \quad (10)$$

λ_2 is a scaling hyperparameter for the heuristic.

Intuitively, this lookahead heuristic brings two benefits. When y_t is a token that would satisfy a multi-token constraint, the lookahead incorporates the score of the *full* constraint. When y_t is a token that is not part of a constraint, the lookahead allows for incorporating the score of a future constraint that would be satisfied if y_t was selected.

We add our lookahead heuristic to the NEUROLOGIC scoring function (Equation 8), and call the resulting decoding procedure NEUROLOGIC A★esque (or, NEUROLOGIC★ in short).

Task	Supervision	Constraints
Commonsense Generation	zero+full	w/
Machine Translation	full	w/
Table-to-text Generation	few	w/
Question Generation	zero	w/
Commonsense Story Generation	full	w/o

Table 1: Tasks and setups considered in this work.

3 Experiments: Constrained Generation

We present experimental results on various constrained generation benchmarks: COMMONGEN (§3.1), constrained machine translation (§3.2), table-to-text generation (§3.3), and interrogative sentence generation (§3.4). NEUROLOGIC★ consistently outperforms NEUROLOGIC and all previous approaches. The improvement is especially substantial in zero-shot and few-shot cases where the search problem is much harder.

Experimental setups. We explore a variety of experimental setups (Table 1). In terms of *supervision*, we consider different configurations of zero-shot, few-shot and full-shot. The former two supervision regimes are particularly important as many realistic generation application do not come with many manually-annotated labeled data. Additionally, we study both *constrained* and *unconstrained* tasks, even though we focus on the former.

Evaluation metrics. We use the following automatic metrics that are commonly used for evaluating text generation: BLEU (Papineni et al., 2002), ROUGE (Lin, 2004), METEOR (Banerjee and Lavie, 2005), CIDEr (Vedantam et al., 2015), SPICE (Anderson et al., 2016) and NIST (Lin and Hovy, 2003). Any other domain specific metrics are detailed in each task description.

3.1 Constrained Commonsense Generation

COMMONGEN (Lin et al., 2020) is a constrained commonsense generation task with lexical constraints.

Decode Method	Automatic Evaluation						Human Evaluation			
	ROUGE-L	BLEU-4	METEOR	CIDEr	SPICE	Coverage	Quality	Plausibility	Concepts	Overall
Supervised										
CBS (Anderson et al., 2017)	38.8	20.6	28.5	12.9	27.1	97.6	2.27	2.35	2.51	2.23
GBS (Hokamp and Liu, 2017)	38.2	18.4	26.7	11.7	26.1	97.4	2.06	2.17	2.29	2.01
DBA (Post and Vilar, 2018a)	38.3	18.7	27.7	12.4	26.3	97.5	2.23	2.30	2.43	2.15
NEUROLOGIC (Lu et al., 2021)	42.8	26.7	30.2	14.7	30.3	<u>97.7</u>	2.54	2.56	2.67	2.50
NEUROLOGIC★ (greedy)	<u>43.6</u>	<u>28.2</u>	**30.8**	15.2	<u>30.8</u>	**97.8**	2.66	<u>2.67</u>	2.73	2.59
NEUROLOGIC★ (sample)	43.4	27.9	**30.8**	<u>15.3</u>	**31.0**	<u>97.7</u>	2.64	2.64	2.74	2.58
NEUROLOGIC★ (beam)	43.2	<u>28.2</u>	<u>30.7</u>	15.2	**31.0**	97.6	<u>2.68</u>	<u>2.67</u>	<u>2.76</u>	<u>2.60</u>
Unsupervised										
TSMH (Zhang et al., 2020)	24.7	2.2	14.5	3.6	15.4	71.5	1.85	1.92	1.95	1.63
NEUROLOGIC (Lu et al., 2021)	41.9	24.7	29.5	14.4	27.5	96.7	2.64	2.52	2.68	2.50
NEUROLOGIC★ (greedy)	**44.3**	**28.6**	<u>30.7</u>	**15.6**	29.6	97.1	**2.78**	**2.70**	**2.77**	**2.70**

Table 2: Performance of various decoding methods with *supervised* or *off-the-shelf* GPT-2 on the COMMONGEN test set, measured with automatic and human evaluations. We only tried NEUROLOGIC★ (greedy) in the unsupervised setting because of the computational cost. The best numbers are **bolded** and the second best ones are <u>underlined</u>.

Words	Method	Generation
cut piece use wood	GBS DBA NEUROLOGIC NEUROLOGIC★	Cut a piece of wood to use as a fence. Cut a piece of wood to use as a fence. Piece of wood used for cutting. **A man cuts a piece of wood using a circular saw.**
ball dog mouth run	GBS DBA NEUROLOGIC NEUROLOGIC★	A dog is run over by a ball and mouth agape. A dog is run over by a ball and bites his mouth. A dog is running and chewing on a ball in its mouth. **A dog running with a ball in its mouth.**
dog scrub soap water	GBS DBA NEUROLOGIC NEUROLOGIC★	Soap and water scrubbed dog with a towel. Soap and water on a dog and scrubbed skin. A dog is scrubbing his paws with soap and water. **A man is scrubbing a dog with soap and water.**

Table 3: Example generations for the COMMONGEN task across supervised NEUROLOGIC★and baselines, including GBS (Hokamp and Liu, 2017), DBA (Post and Vilar, 2018a), and NEUROLOGIC (Lu et al., 2021)

Given a set of concepts (*e.g.*, {throw, run, javelin, track}), the task is to generate a coherent sentence describing a plausible scenario using all of the given concepts (*e.g.*, "a man runs on a track and throws a javelin.").

Approach and Baselines. Following Lu et al. (2021), we enforce that each given concept c_i must appear in output $\mathbf{y}$ under some morphological inflection. We experiment with both supervised and zero-shot settings. In the supervised setting, we formulate it as conditional sentence generation task and finetune GPT-2 (Radford et al., 2019) as a sequence-to-sequence model. In the zero-shot setting, we use GPT-2 off-the-shelf (no fine-tuning), and rely on constrained decoding to guide the generations. We compare with previous constrained decoding algorithms, including CBS (Anderson et al., 2017), GBS (Hokamp and Liu, 2017), DBA (Post and Vilar, 2018a), NEUROLOGIC (Lu et al., 2021) and TSMH (Zhang et al., 2020)

Metrics Following Lin et al. (2020), we report automatic generation metrics as well as *coverage*, defined as the average percentage of the provided concepts that are present in lemmatized outputs. Additionally, we conduct human evaluation on 100 test examples with workers from Amazon Mechanical Turk (AMT). We include our evaluation template in Figure 5 of Appendix A. Workers are given a pair of concepts and a model generation, and asked to rate each pair on *language quality*, *scenario plausibility*, *coverage of given concepts*, and an *overall score*, in the Likert scale: *Agree*, *Neutral*, and *Disagree*. Each pair is rated by 3 workers.

Results. Table 2 compares different constrained decoding methods on top of the finetuned and off-the-shelf GPT-2, in supervised and zero-shot settings respectively. The key observations are:

1. NEUROLOGIC★ outperforms all previous constrained-decoding methods in both supervised and zero-shot settings. Surprisingly, *unsupervised* NEUROLOGIC★ outperforms all supervised methods based on human evaluation.
2. Compared to vanilla NEUROLOGIC, NEUROLOGIC★ improves the generation quality while maintaining high constraint satisfaction. The difference is especially substantial in the zero-shot case, where there is more room for incorporating constraint-driven signals due to the lack of supervision and the large output space.
3. NEUROLOGIC★ reaches similar performance with different lookahead strategies, among which beam lookahead slightly outperforms the

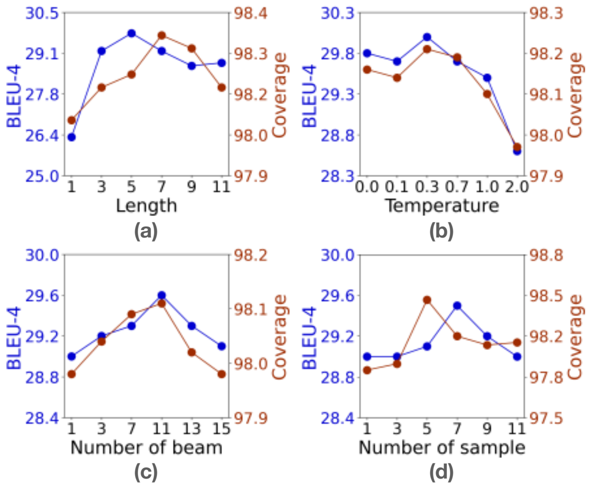


Figure 2: Performance (y-axis) of supervised GPT-2 in terms of **BLEU-4** and **Coverage** with varying lookahead parameters (x-axis) on COMMONGEN validation set.

others based on human evaluation, and greedy lookahead has the lowest runtime.

Studying lookahead strategies. With an infinite lookahead length ℓ and number of lookaheads $|\mathcal{L}_\ell|$, lookahead decoding exactly solves Equation 3. For practical choices of ℓ and $|\mathcal{L}_\ell|$, we empirically study how varying the lookahead strategy and hyperparameters affects performance. In Figure 2, we study the greedy, soft, beam, and sampling lookahead strategies (§2.1).

Figure 2(a) shows the effect of increasing the lookahead horizon ℓ for the greedy strategy. Increasing the horizon improves up to one point – e.g., 5-7 steps – then decreases thereafter, likely due to the difficulty of long-horizon approximation.

Figure 2(b) studies the temperature in the soft lookahead, showing that greedy ($\tau = 0.0$) performs well, with slight gains if τ is carefully selected. The results suggest that one can safely bypass tuning τ using fast, greedy lookahead.

Next, Figure 2(c) shows that with beam lookahead, increasing the beam width improves performance up to a certain point (here, 11). Similarly, increasing the number of samples with sampling lookahead improves over a single sample, and then reaches an inflection point (Figure 2(d)).

3.2 Constrained Machine Translation

It is often critical to have control over machine translation output. For example, domain-specific dictionaries can be incorporated to force a model

Method	Dinu et al. BLUE	Dinu et al. Term%	Marian MT BLUE	Marian MT Term%
Unconstrained	25.8	76.3	32.9	85.0
train-by-app.	26.0	92.9	–	–
train-by-rep.	26.0	94.5	–	–
Post and Vilar (2018a)	25.3	82.0	33.0	94.3
NEUROLOGIC	26.5	95.1	33.4	<u>97.1</u>
NEUROLOGIC★ (greedy)	**26.7**	**95.8**	**33.7**	**97.2**
NEUROLOGIC★ (sample)	<u>26.6</u>	<u>95.4</u>	**33.7**	**97.2**
NEUROLOGIC★ (beam)	<u>26.6</u>	**95.8**	<u>33.6</u>	**97.2**

Table 4: Results on constrained machine translation. The left section uses the same two-layer transformer model as Dinu et al. (2019) for fair comparisons. The right one decodes a stronger Marian MT EN-DE model. The highlighted methods modify training data specifically for constrained decoding, and thus cannot be applied to off-the-shelf models. The best numbers are **bolded** and the second best ones are <u>underlined</u>.

# T	# Sents.	Decode Method	BLEU	Term%
1	378	Beam search	25.4	79.6
		NEUROLOGIC	<u>26.2</u>	<u>95.2</u>
		NEUROLOGIC★	**26.3**	**95.8**
2+	36	Beam search	28.1	85.0
		NEUROLOGIC	<u>28.9</u>	<u>93.7</u>
		NEUROLOGIC★	**29.3**	**96.5**

Table 5: Constrained Machine Translation performance broken down by the number of constraint terms (# T). All configurations use the two-layer tranformer from Dinu et al. (2019). The best numbers are **bolded** and the second best ones are <u>underlined</u>.

to use certain terminology (Post and Vilar, 2018a; Dinu et al., 2019). To achieve this goal, much recent work proposed constrained decoding algorithms (Chatterjee et al., 2017; Hokamp and Liu, 2017; Hasler et al., 2018; Hu et al., 2019, *inter alia*) or specialized training (Dinu et al., 2019). We demonstrate that NEUROLOGIC★ can be readily applied to off-the-shelf MT systems for constrained machine translation. Specifically, we follow the setup in Dinu et al. (2019) and evaluate our method on the WMT17 EN-DE test data (Bojar et al., 2017). The constraint here is to integrate a given custom terminology into the translation output; constraint terms are automatically created from the IATE EU terminology database for 414 test sentences.[2]

Approach, Baselines, and Metrics. We experiment with two MT systems: Dinu et al. (two-layer transformer) and the off-the-shelf Marian MT (Junczys-Dowmunt et al., 2018). We compare with previous constrained decoding algorithms, including DBA (Post and Vilar, 2018a), NEUROLOGIC

[2] `https://github.com/mtresearcher/terminology_dataset`.

(Lu et al., 2021) and also specialized training proposed by Dinu et al. (2019). Following Dinu et al. (2019), we report BLEU scores and term use rates, computed as the percentage of times a given constraint term was generated in the output out of the total number of constraint terms.

Results. Table 4 presents experimental results with Dinu et al.'s model and Marian MT. We can see that in either case, NEUROLOGIC★ outperforms all prior methods both in BLEU and term coverage. Besides better generation quality and constraint coverage, NEUROLOGIC★ also benefits from its plug-and-play flexibility with any off-the-shelf MT system compared to previous training-based methods. Table 5 breaks down the model performance by the number of constraint terms. We see that NEUROLOGIC★ improves upon the others, especially when the constraint is complex with multiple constraint terms. (*e.g.*, 96.5 vs. 93.7 from NEUROLOGIC in term coverage).

3.3 Table-to-text Generation

The table-to-text task aims to generate natural language text conditioned on structured table data; their applications include automatic generation of weather/sports reports (Liang et al., 2009; Wiseman et al., 2017) or dialogue responses (Wen et al., 2016). Constrained generation algorithms can be used to ensure that the output text is consistent with the input structured data. We follow the few-shot setup of Chen et al. (2020b) on the E2ENLG (Dušek et al., 2018) dataset, where we use randomly-sampled 0.1%, 0.5%, 1%, 5% of training instances for finetuning.

Approach, Baselines, and Metrics. Following Shen et al. (2019), we linearize the given table into a string and finetune GPT-2 with given few-shot examples. We first compare NEUROLOGIC★ with three previous constrained decoding algorithms: CBS (Anderson et al., 2017), GBS (Hokamp and Liu, 2017), and NEUROLOGIC (Lu et al., 2021), based on few-shot GPT-2 finetuned with 0.1% data. Then we compare our approach, NEUROLOGIC★ on top of GPT-2, with previous table-to-text methods, including TGen (Dušek and Jurčíček, 2016), Template-GPT-2 (Chen et al., 2020a), KGPT (Chen et al., 2020b), in multiple few-shot settings with various numbers of training instances. We report standard automatic metrics used in the E2ENLG challenge, as well as information coverage– the average percentage of given information that is present in the generation.

Decode Method	NIST	BLEU	METEOR	CIDEr	ROUGE	Coverage
Beam Search	3.82	42.8	32.6	10.8	57.8	73.6
CBS	6.50	42.3	36.4	13.0	54.3	91.6
GBS	6.26	40.7	36.7	12.9	54.2	94.1
NEUROLOGIC	6.95	47.6	38.9	16.3	58.7	97.6
NEUROLOGIC★ (greedy)	**7.11**	<u>49.2</u>	<u>40.0</u>	**17.5**	**60.0**	**100.0**
NEUROLOGIC★ (beam)	<u>7.01</u>	48.9	<u>40.0</u>	<u>17.2</u>	<u>59.8</u>	<u>99.9</u>
NEUROLOGIC★ (sample)	**7.11**	**49.3**	**40.1**	**17.5**	**60.0**	**100.0**

Table 6: Performance of different decoding methods with few-shot GPT-2 finetuned on 0.1% E2ENLG data. The best numbers are **bolded** and the second best ones are <u>underlined</u>.

Method	0.1%	0.5%	1%	5%
TGen (Dušek and Jurčíček, 2016)	3.6	27.9	35.2	57.3
Template-GPT-2 (Chen et al., 2020a)	22.5	47.8	53.3	59.9
KGPT-Graph (Chen et al., 2020b)	39.8	53.3	55.1	61.5
KGPT-Seq (Chen et al., 2020b)	40.2	53.0	54.1	61.1
GPT-2	42.8	<u>57.1</u>	56.8	61.1
GPT-2 + NEUROLOGIC	<u>47.6</u>	56.9	<u>58.0</u>	<u>62.9</u>
GPT-2 + NEUROLOGIC★ (greedy)	**49.2**	**58.0**	**58.4**	**63.4**

Table 7: Few-shot results (BLEU-4) on E2ENLG test set with 0.1%, 0.5%, 1%, 5% of training instances. The best numbers are **bolded** and the second best ones are <u>underlined</u>.

Results. Table 6 presents results from varying decoding algorithms based on few-shot GPT-2 finetuned with 0.1% of the data. NEUROLOGIC★ substantially outperforms all previous methods with respect to all metrics; it consistently improves generation quality while achieving (almost) perfect constraint satisfaction. Previous work, like CBS and GBS, improves constraint satisfaction, but negatively affects the text quality, as indicated by drops in BLEU and ROUGE. Table 7 compares NEUROLOGIC★ on top of GPT-2 with previous table-to-text approaches. As before, NEUROLOGIC★ outperforms all prior approaches by a large margin, even if the latter ones leverage either specialized model architecture or additional pretraining on massive table-to-text corpora. Additionally, Figure 3 compares the performance (y-axis) of few-shot GPT-2 with NEUROLOGIC★ (**purple line**), NEUROLOGIC (**blue line**), and conventional beam search (**black line**) as a function of the varying amount of training instances (x-axis). We find the relative gain brought by NEUROLOGIC★ increases as we reduce the amount of few-shot examples. Results above demonstrate the promise of decoding algorithms to address unsatisfying performance in few-shot scenarios due to insufficient learning.

Decode Method	Automatic Evaluation						Human Evaluation			
	ROUGE	BLEU	METEOR	CIDEr	SPICE	Coverage	Grammar	Fluency	Meaningfulness	Overall
CGMH (Miao et al., 2019)	28.8	2.0	18.0	5.5	21.5	18.3	2.28	2.34	2.11	2.02
TSMH (Zhang et al., 2020)	42.0	4.3	25.9	10.4	37.7	92.7	2.35	2.28	2.37	2.22
NEUROLOGIC (Lu et al., 2021)	38.8	11.2	24.5	18.0	41.7	90.6	2.78	2.71	2.49	2.51
NEUROLOGIC★ (greedy)	**43.7**	**14.7**	28.0	**20.9**	47.7	**100.0**	**2.83**	2.77	2.74	**2.76**
NEUROLOGIC★ (beam)	42.9	14.4	27.8	20.3	46.9	**100.0**	2.81	**2.86**	**2.76**	2.75
NEUROLOGIC★ (sample)	43.5	14.6	**28.2**	20.8	**47.8**	**100.0**	**2.83**	2.75	**2.76**	2.73

Table 8: Performance of different unsupervised decoding algorithms on interrogative question generation.

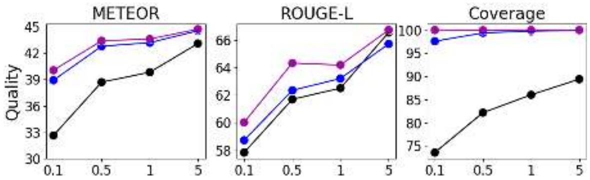


Figure 3: Performance (y-axis) of supervised GPT-2 on E2ENLG, with a varying amount of training data for supervision (x-axis). The **purple**, **blue**, and **black** line denote decoding with NEUROLOGIC★, NEUROLOGIC and conventional beam search respectively.

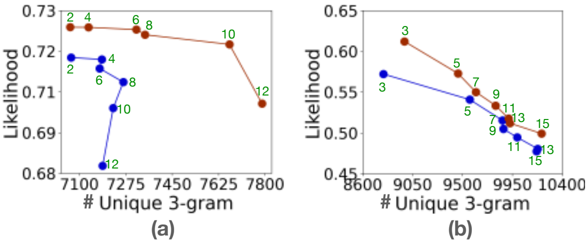


Figure 4: Likelihood (y-axis) vs. number of unique 3-grams (x-axis) using supervised GPT-2 on RocStories. Figure **(a)** denotes decoding with beam search, with a varying amount of beam size. Figure **(b)** denotes decoding with top-k sampling, with a varying amount of k value. The **brown** and **blue** line denotes with and without A★esque heuristics separately.

3.4 Constrained Question Generation

Despite the success of supervised techniques in natural language generation, it needs to be trained with massive task-specific data, which is non-trivial to acquire. We investigate a zero-shot text generation task proposed by Zhang et al. (2020): constrained question generation, where no training data is available. Given a set of keywords (e.g., Nevada, desert, border), the task is to use an off-the-self language model to generate an interrogative question containing given keywords (e.g., "What is the name of the desert near the border of Nevada?"). Two types of constraints are enforced for this task: 1) keyword constraints - the output question must include all the keywords provided, and 2) syntactic constraints - the output question must be in the interrogative form, the first word must be *wh-* question words, and the second or third word must be auxiliary verbs or copula words.

Approach, Baselines, and Metrics. We leverage off-the-shelf language model GPT-2 and compare NEUROLOGIC★ with three previous constrained decoding methods, CGMH (Miao et al., 2019), TSMH (Zhang et al., 2020) and NEUROLOGIC (Lu et al., 2021). CGMH and TSMH are two Metropolis-Hastings sampling-based decoding algorithms that have shown strong performance in unsupervised constrained generation. For automatic evaluation, we report standard generation metrics and keyword Coverage similar to previous task COMMONGEN. For the human evaluation, we sample 100 test examples and employ workers from AMT to evaluate the generated interrogative questions. Workers are given a set of keywords and model generation. They are asked to evaluate the generation based on 3 individual qualities (*i.e.*, grammar, fluency, meaningfulness) and provide an overall quality score, using the 3-point Likert scale. Each example is averaged across 3 workers. We include the human evaluation template in Figure 6 of the Appendix A.

Results. Table 8 presents comparisons across different decoding methods based on off-the-shelf language models. We can see that NEUROLOGIC★ outperforms all previous methods with respect to both automatic and manual metrics; it remarkably enhances the generation quality while achieves perfect constraint satisfaction. The difference between NEUROLOGIC and NEUROLOGIC★ is particularly large compared to other tasks. The search problem is much harder here, due to the lack of supervision and complex logical constraint involving both keywords and syntax. Results above demonstrate the effectiveness of NEUROLOGIC★ in tackling more challenging constrained generation problems.

Decode Method	Fluency			Diversity		Human Eval				
	PPL	BLEU-1	BLEU-2	Uniq. 3-gram	Uniq. 4-gram	Grammar	Fluency	Coherence	Interest	Overall
beam search	2.24	33.7	16.5	34.09k	41.91k	2.81	2.50	2.46	2.27	2.32
beam search + A★esque (greedy)	**2.11**	<u>34.3</u>	<u>16.7</u>	34.94k	43.02k	**2.94**	<u>2.71</u>	2.56	2.50	<u>2.57</u>
beam search + A★esque (beam)	<u>2.14</u>	**34.4**	**16.8**	<u>35.03k</u>	<u>43.12k</u>	**2.94**	**2.72**	**2.62**	**2.61**	**2.63**
beam search + A★esque (sample)	2.16	**34.4**	<u>16.7</u>	**35.41k**	**43.64k**	<u>2.92</u>	<u>2.71</u>	<u>2.59</u>	<u>2.52</u>	<u>2.57</u>
top-k sample	4.01	31.4	13.9	<u>48.36k</u>	<u>56.62k</u>	2.69	2.38	2.23	2.30	2.15
top-k sample + A★esque (greedy)	**3.68**	<u>32.1</u>	<u>14.3</u>	**48.44k**	**56.63k**	**2.88**	**2.57**	**2.48**	**2.49**	**2.47**
top-k sample + A★esque (beam)	3.75	**32.2**	**14.4**	48.27k	56.36k	<u>2.84</u>	2.49	2.39	2.40	2.34
top-k sample + A★esque (sample)	<u>3.70</u>	32.0	14.2	48.04k	56.15k	<u>2.84</u>	<u>2.55</u>	<u>2.47</u>	<u>2.48</u>	<u>2.44</u>

Table 9: Performance of different decoding algorithms on RocStories test set.

4 Experiments: Unconstrained Generation

So far we have experimented with constrained text generation, but here we demonstrate that NEUROLOGIC★ decoding can also improve *unconstrained* generation. Specifically, we investigate whether A★esque decoding with our unconstrained lookahead heuristic (Equation 7) can (i) improve beam search, which typically struggles in open-ended settings (Holtzman et al., 2020; Welleck et al., 2019b), (ii) improve *sampling* algorithms that are commonly used in open-ended generation.

4.1 Commonsense Story Generation

We investigate story generation with RocStories (Mostafazadeh et al., 2016). Given the first sentence as a prompt $\mathbf{x}$, the task is to generate the rest of story continuation $\mathbf{y}$.

Approach, Baselines and Metrics. We consider storytelling as a conditional generation task, and finetune GPT-2 as a sequence-to-sequence model.

We apply A★esque decoding with our unconstrained lookahead heuristic (Equation 7) to (i) beam search, the setting used so far in the experiments, and (ii) top-k sampling (Fan et al., 2018), a commonly used sampling algorithm in open-ended generation. For top-k sampling, we use the heuristic to adjust the probability scores, then renormalize.

For automatic evaluation, besides commonly used automatic metrics for storytelling, including perplexity and BLEU, we also report unique n-grams as a measure for diversity. For the human evaluation, we sample 100 stories from the test set and we employ workers from AMT to evaluate the model generations. Workers are given the first sentence of the story (i.e., prompt), and the model-generated continuation of the story. They are asked to evaluate the continuation of the story on 4 individual qualities (i.e., grammar, fluency, story flow, interestingness) and provide an overall quality score, using the 3-point Likert scale. Each example is averaged across 3 workers. We include the human evaluation template in Figure 7 of the Appendix A.

Results. Table 9 presents the results of beam search and top-k sampling with and without A★esque heuristics. We can see that A★esque heuristics enable both beam search and top-k sampling to generate more fluent, coherent and interesting stories. For beam search, our A★esque heuristic not only enhances generation quality– e.g. improving human evaluation scores from 2.32 to 2.63– but also boosts generation diversity, as reflected by the number of unique n-grams. For top-k sampling, A* heuristics also improves generation quality, while maintaining comparable diversity. We notice that beam lookahead works the best for beam search, and greedy lookahead works the best for top-k sampling. We suspect that beam lookahead gives the most accurate estimate of the future path that beam search is likely to reach, while the greedy lookahead provides an estimate that is lower than what obtained by beam search, which may better resemble a continuation from top-k sampling.

Ablations. We study the effect of A★esque decoding with different decoding hyperparameters: beam size in beam search and k value in top-k sampling. Figure 4 plots the fluency (measured by likelihood) versus diversity (measured by unique 3-grams) for generations with various beam sizes or k values. Ideally, we want generations to be both fluent and diverse, centering around the top-right center. However, we observe a fluency and diversity tradeoff in practice. Interestingly, we observe that A★esque decoding flattens this trend and results in larger area under the curve. The effect is especially obvious for beam search. The results above demonstrate that A★esque decoding can guide generation towards a more favorable output space that cannot be reached with conventional decoding methods, regardless of decoding hyperparameters.

5 Related Work

A* search in NLP. Many classical NLP problems (*e.g.*, parsing, text alignment) can be seen as structured prediction subject to a set of task-specific constraints. For many such problems, A* search has been used effectively (Och et al., 2001; Haghighi et al., 2007; Hopkins and Langmead, 2009; Meister et al., 2020). For example, Klein and Manning (2003); Zhang and Gildea (2006); Auli and Lopez (2011); Lee et al. (2016) have used it in the context of parsing. Similar approaches are used for finding high-probability alignments (Naim et al., 2013). Despite these applications, applying informed heuristic search to text generation with autoregressive language models has been under-explored, which is the focus of this work.

Decoding strategies for text generation. The rise of autoregressive language models like GPT (Radford et al., 2018) has inspired a flurry of work on decoding strategies (Post and Vilar, 2018a; Ippolito et al., 2019; Zheng et al., 2020; Leblond et al., 2021; West et al., 2021). These works often focus on incorporating factors like diversity (Ippolito et al., 2019), fluency (Holtzman et al., 2020) or constraints (Anderson et al., 2017; Hokamp and Liu, 2017; Post and Vilar, 2018b; Miao et al., 2019; Welleck et al., 2019a; Zhang et al., 2020; Qin et al., 2020; Lu et al., 2021). Among constrained decoding methods, previous works such as constrained beam search (Anderson et al., 2017) and grid beam search (Hokamp and Liu, 2017), have worked on extending beam search to satisfy lexical constraints during generation.

Other works have focused on the mismatch between monotonic decoding and satisfying constraints that may depend on a full generation. Miao et al. (2019) propose a sampling-based conditional generation method using Metropolis-Hastings sampling (CGMH), where the constrained words are inserted/deleted/edited by the Metropolis-Hastings scheme, allowing a full generation to be edited towards desired properties. Welleck et al. (2019a) develop a tree-based constrained text generation, which recursively generates text in a non-monotonic order given constraint tokens, ensuring constraints are satisfied. Zhang et al. (2020) proposes tree search enhanced MCMC that handles combinatorial constraints (TSMH). Qin et al. (2020) instead casts constrained decoding as a continuous optimization problem that permits gradient-based updates. West et al. (2021) encodes constraints as generated contexts which models condition on to encourage satisfaction. Compared to these past works, NeuroLogic A★esque explicitly samples future text to estimate viability of different paths towards satisfying constraints. Our approach is based on Lu et al. (2021), which incorporates constraints in Conjunctive Normal Form (CNF), but we extend this into the future with our lookahead heuristics.

6 Conclusion

Inspired by the A* search algorithm, we introduce NeuroLogic A★esque decoding, which brings A*-like heuristic estimates of the *future* to common *left-to-right* decoding algorithms for neural text generation. NeuroLogic A★esque's lookahead heuristics improve over existing decoding methods (*e.g.*, NeuroLogic, beam, greedy, sample decoding methods) in both *constrained* and *unconstrained* settings across a wide spectrum of tasks. Our work demonstrates the promise of moving beyond the current paradigm of unidirectional decoding for text generation, by taking bidirectional information from both the *past* and *future* into account to generate more globally compatible text.

Acknowledgment

This work was supported in part by Natural Sciences and Engineering Research Council of Canada (NSERC) (funding reference number 401233309), DARPA MCS program through NIWC Pacific (N66001-19-2-4031), Google Cloud Compute, and Allen Institute for AI, Microsoft PhD Fellowship.

References

Peter Anderson, Basura Fernando, Mark Johnson, and Stephen Gould. 2016. Spice: Semantic propositional image caption evaluation. In *European conference on computer vision*, pages 382–398. Springer.

Peter Anderson, Basura Fernando, Mark Johnson, and Stephen Gould. 2017. Guided open vocabulary image captioning with constrained beam search. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 936–945, Copenhagen, Denmark. Association for Computational Linguistics.

Michael Auli and Adam Lopez. 2011. Efficient CCG parsing: A* versus adaptive supertagging. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human*

Language Technologies, pages 1577–1585, Portland, Oregon, USA. Association for Computational Linguistics.

Satanjeev Banerjee and Alon Lavie. 2005. Meteor: An automatic metric for mt evaluation with improved correlation with human judgments. In *Proceedings of the acl workshop on intrinsic and extrinsic evaluation measures for machine translation and/or summarization*, pages 65–72.

Ondřej Bojar, Rajen Chatterjee, Christian Federmann, Yvette Graham, Barry Haddow, Shujian Huang, Matthias Huck, Philipp Koehn, Qun Liu, Varvara Logacheva, Christof Monz, Matteo Negri, Matt Post, Raphael Rubino, Lucia Specia, and Marco Turchi. 2017. Findings of the 2017 conference on machine translation (WMT17). In *Proceedings of the Second Conference on Machine Translation*, pages 169–214, Copenhagen, Denmark. Association for Computational Linguistics.

T. Brown, B. Mann, Nick Ryder, Melanie Subbiah, J. Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, G. Krüger, T. Henighan, R. Child, Aditya Ramesh, D. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, E. Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, J. Clark, Christopher Berner, Sam McCandlish, A. Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. In *Advances in Neural Information Processing Systems (NeurIPS)*.

Rajen Chatterjee, Matteo Negri, Marco Turchi, Marcello Federico, Lucia Specia, and Frédéric Blain. 2017. Guiding neural machine translation decoding with external knowledge. In *Proceedings of the Second Conference on Machine Translation*, pages 157–168, Copenhagen, Denmark. Association for Computational Linguistics.

Wenhu Chen, Jianshu Chen, Yu Su, Zhiyu Chen, and William Yang Wang. 2020a. Logical natural language generation from open-domain tables. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7929–7942, Online. Association for Computational Linguistics.

Wenhu Chen, Yu Su, Xifeng Yan, and William Yang Wang. 2020b. KGPT: Knowledge-grounded pretraining for data-to-text generation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 8635–8648, Online. Association for Computational Linguistics.

Yining Chen, Sorcha Gilroy, Andreas Maletti, Jonathan May, and Kevin Knight. 2018. Recurrent neural networks as weighted language recognizers. 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 2261–2271, New Orleans, Louisiana. Association for Computational Linguistics.

Sumanth Dathathri, Andrea Madotto, Janice Lan, Jane Hung, Eric Frank, Piero Molino, Jason Yosinski, and Rosanne Liu. 2019. Plug and play language models: A simple approach to controlled text generation. In *International Conference on Learning Representations*.

Georgiana Dinu, Prashant Mathur, Marcello Federico, and Yaser Al-Onaizan. 2019. Training neural machine translation to apply terminology constraints. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3063–3068, Florence, Italy. Association for Computational Linguistics.

Ondřej Dušek and Filip Jurčíček. 2016. Sequence-to-sequence generation for spoken dialogue via deep syntax trees and strings. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 45–51, Berlin, Germany. Association for Computational Linguistics.

Ondřej Dušek, Jekaterina Novikova, and Verena Rieser. 2018. Findings of the E2E NLG Challenge. In *Proc. of the 11th International Conference on Natural Language Generation*, pages 322–328, Tilburg, The Netherlands. Association for Computational Linguistics.

Angela Fan, Mike Lewis, and Yann Dauphin. 2018. Hierarchical neural story generation. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 889–898, Melbourne, Australia. Association for Computational Linguistics.

Aria Haghighi, John DeNero, and Dan Klein. 2007. Approximate factoring for A* search. In *Human Language Technologies 2007: The Conference of the North American Chapter of the Association for Computational Linguistics; Proceedings of the Main Conference*, pages 412–419, Rochester, New York. Association for Computational Linguistics.

Peter E. Hart, Nils J. Nilsson, and Bertram Raphael. 1968. A formal basis for the heuristic determination of minimum cost paths. *IEEE Transactions on Systems Science and Cybernetics*, 4(2):100–107.

Eva Hasler, Adrià de Gispert, Gonzalo Iglesias, and Bill Byrne. 2018. Neural machine translation decoding with terminology constraints. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 506–512, New Orleans, Louisiana. Association for Computational Linguistics.

Chris Hokamp and Qun Liu. 2017. Lexically constrained decoding for sequence generation using grid beam search. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1535–1546, Vancouver, Canada. Association for Computational Linguistics.

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

Mark Hopkins and Greg Langmead. 2009. Cube pruning as heuristic search. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing*, pages 62–71.

J. Edward Hu, Huda Khayrallah, Ryan Culkin, Patrick Xia, Tongfei Chen, Matt Post, and Benjamin Van Durme. 2019. Improved lexically constrained decoding for translation and monolingual rewriting. 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 839–850, Minneapolis, Minnesota. Association for Computational Linguistics.

Zhiting Hu, Zichao Yang, Xiaodan Liang, Ruslan Salakhutdinov, and Eric P Xing. 2017. Toward controlled generation of text. In *International Conference on Machine Learning*, pages 1587–1596. PMLR.

Daphne Ippolito, Reno Kriz, João Sedoc, Maria Kustikova, and Chris Callison-Burch. 2019. Comparison of diverse decoding methods from conditional language models. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3752–3762.

Marcin Junczys-Dowmunt, Roman Grundkiewicz, Tomasz Dwojak, Hieu Hoang, Kenneth Heafield, Tom Neckermann, Frank Seide, Ulrich Germann, Alham Fikri Aji, Nikolay Bogoychev, André F. T. Martins, and Alexandra Birch. 2018. Marian: Fast neural machine translation in C++. In *Proceedings of ACL 2018, System Demonstrations*, pages 116–121, Melbourne, Australia. Association for Computational Linguistics.

Dan Klein and Christopher D. Manning. 2003. A* parsing: Fast exact Viterbi parse selection. In *Proceedings of the 2003 Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics*, pages 119–126.

Richard E Korf. 1985. Depth-first iterative-deepening: An optimal admissible tree search. *Artificial intelligence*, 27(1):97–109.

Rémi Leblond, Jean-Baptiste Alayrac, Laurent Sifre, Miruna Pislar, Jean-Baptiste Lespiau, Ioannis Antonoglou, Karen Simonyan, and Oriol Vinyals. 2021. Machine translation decoding beyond beam search. *arXiv preprint arXiv:2104.05336*.

Kenton Lee, Mike Lewis, and Luke Zettlemoyer. 2016. Global neural CCG parsing with optimality guarantees. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2366–2376, Austin, Texas. Association for Computational Linguistics.

Percy Liang, Michael Jordan, and Dan Klein. 2009. Learning semantic correspondences with less supervision. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP*.

Bill Yuchen Lin, Ming Shen, Wangchunshu Zhou, Pei Zhou, Chandra Bhagavatula, Yejin Choi, and Xiang Ren. 2020. Commongen: A constrained text generation challenge for generative commonsense reasoning. In *Findings of EMNLP*.

Chin-Yew Lin. 2004. Rouge: A package for automatic evaluation of summaries. *Text Summarization Branches Out*.

Chin-Yew Lin and Eduard Hovy. 2003. Automatic evaluation of summaries using n-gram cooccurrence statistics. In *Proceedings of the 2003 Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics*, pages 150–157.

Ximing Lu, Peter West, Rowan Zellers, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. 2021. NeuroLogic decoding: (un)supervised neural text generation with predicate logic constraints. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4288–4299, Online. Association for Computational Linguistics.

Clara Meister, Tim Vieira, and Ryan Cotterell. 2020. Best-first beam search. *Transactions of the Association for Computational Linguistics*, 8:795–809.

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

Nasrin Mostafazadeh, Nathanael Chambers, Xiaodong He, Devi Parikh, Dhruv Batra, Lucy Vanderwende, Pushmeet Kohli, and James Allen. 2016. A corpus and cloze evaluation for deeper understanding of commonsense stories. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 839–849, San Diego, California. Association for Computational Linguistics.

Iftekhar Naim, Daniel Gildea, Walter Lasecki, and Jeffrey P Bigham. 2013. Text alignment for real-time crowd captioning. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 201–210.

Franz Josef Och, Nicola Ueffing, and Hermann Ney. 2001. An efficient a* search algorithm for statistical machine translation. In *Proceedings of the ACL 2001 Workshop on Data-Driven Methods in Machine Translation*.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. BLEU: a method for automatic evaluation of machine translation. In *ACL*, pages 311–318.

Judea Pearl. 1984. Heuristics - intelligent search strategies for computer problem solving. In *Addison-Wesley series in artificial intelligence*.

Ira Pohl. 1970. *First Results on the Effect of Error in Heuristic Search*.

Matt Post and David Vilar. 2018a. Fast lexically constrained decoding with dynamic beam allocation for neural machine translation. 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 1314–1324, New Orleans, Louisiana. Association for Computational Linguistics.

Matt Post and David Vilar. 2018b. Fast lexically constrained decoding with dynamic beam allocation for neural machine translation. 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 1314–1324.

Lianhui Qin, Vered Shwartz, Peter West, Chandra Bhagavatula, Jena D Hwang, Ronan Le Bras, Antoine Bosselut, and Yejin Choi. 2020. Backpropagation-based decoding for unsupervised counterfactual and abductive reasoning. In *EMNLP*.

Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. 2018. Improving language understanding by generative pre-training.

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

Sheng Shen, Daniel Fried, Jacob Andreas, and Dan Klein. 2019. Pragmatically informative text generation. 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 4060–4067, Minneapolis, Minnesota. Association for Computational Linguistics.

Ramakrishna Vedantam, C Lawrence Zitnick, and Devi Parikh. 2015. Cider: Consensus-based image description evaluation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4566–4575.

Sean Welleck, Kianté Brantley, Hal Daumé Iii, and Kyunghyun Cho. 2019a. Non-monotonic sequential text generation. In *International Conference on Machine Learning*, pages 6716–6726. PMLR.

Sean Welleck, Ilia Kulikov, Stephen Roller, Emily Dinan, Kyunghyun Cho, and Jason Weston. 2019b. Neural text generation with unlikelihood training. In *International Conference on Learning Representations*.

Tsung-Hsien Wen, Milica Gašić, Nikola Mrkšić, Lina M. Rojas-Barahona, Pei-Hao Su, David Vandyke, and Steve Young. 2016. Multi-domain neural network language generation for spoken dialogue systems. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 120–129, San Diego, California. Association for Computational Linguistics.

Peter West, Ximing Lu, Ari Holtzman, Chandra Bhagavatula, Jena D. Hwang, and Yejin Choi. 2021. Reflective decoding: Beyond unidirectional generation with off-the-shelf language models. In *ACL/IJCNLP*.

Sam Wiseman, Stuart Shieber, and Alexander Rush. 2017. Challenges in data-to-document generation. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*.

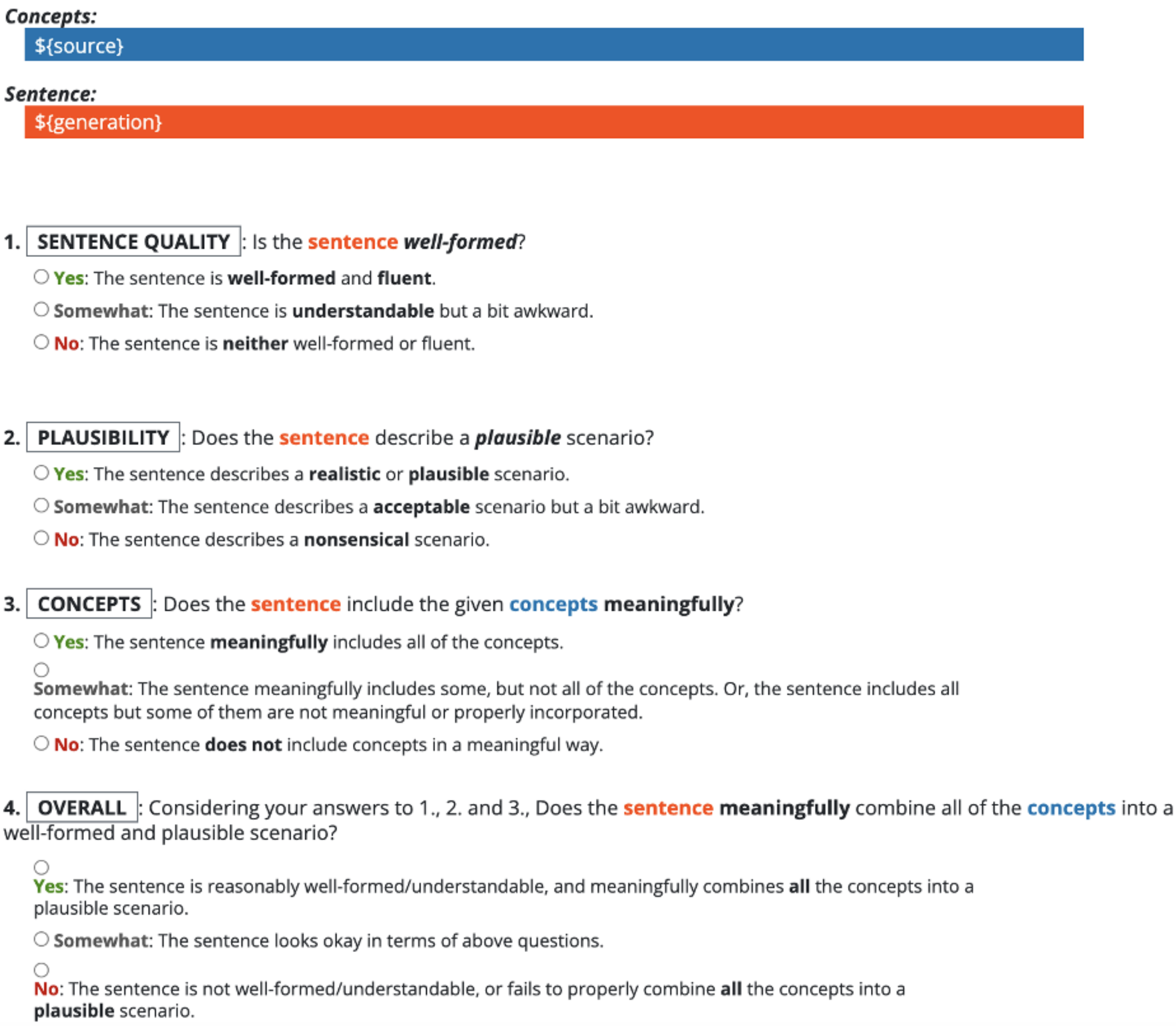
Hao Zhang and Daniel Gildea. 2006. Efficient search for inversion transduction grammar. In *Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing*, pages 224–231.

Maosen Zhang, Nan Jiang, Lei Li, and Yexiang Xue. 2020. Language generation via combinatorial constraint satisfaction: A tree search enhanced Monte-Carlo approach. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 1286–1298, Online. Association for Computational Linguistics.

Renjie Zheng, Mingbo Ma, Baigong Zheng, Kaibo Liu, and Liang Huang. 2020. Opportunistic decoding with timely correction for simultaneous translation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 437–442.

A Human Evaluation

We include screenshots of the human evaluation templates for CommonGen (Figure 5), Interrogative Sentence Generation (Figure 6), and RocStories (Figure 7) tasks.



Concepts:

${source}

Sentence:

${generation}

1. SENTENCE QUALITY : Is the **sentence** ***well-formed***?

○ **Yes**: The sentence is **well-formed** and **fluent**.

○ **Somewhat:** The sentence is **understandable** but a bit awkward.

○ **No**: The sentence is **neither** well-formed or fluent.

2. PLAUSIBILITY : Does the **sentence** describe a ***plausible*** scenario?

○ **Yes**: The sentence describes a **realistic** or **plausible** scenario.

○ **Somewhat**: The sentence describes a **acceptable** scenario but a bit awkward.

○ **No**: The sentence describes a **nonsensical** scenario.

3. CONCEPTS : Does the **sentence** include the given **concepts** **meaningfully**?

○ **Yes**: The sentence **meaningfully** includes all of the concepts.

○
Somewhat: The sentence meaningfully includes some, but not all of the concepts. Or, the sentence includes all concepts but some of them are not meaningful or properly incorporated.

○ **No**: The sentence **does not** include concepts in a meaningful way.

4. OVERALL : Considering your answers to 1., 2. and 3., Does the **sentence** **meaningfully** combine all of the **concepts** into a well-formed and plausible scenario?

○
Yes: The sentence is reasonably well-formed/understandable, and meaningfully combines **all** the concepts into a plausible scenario.

○ **Somewhat**: The sentence looks okay in terms of above questions.

○
No: The sentence is not well-formed/understandable, or fails to properly combine **all** the concepts into a **plausible** scenario.

Figure 5: Human evaluation template for the Constrained Commonsense Generation task.

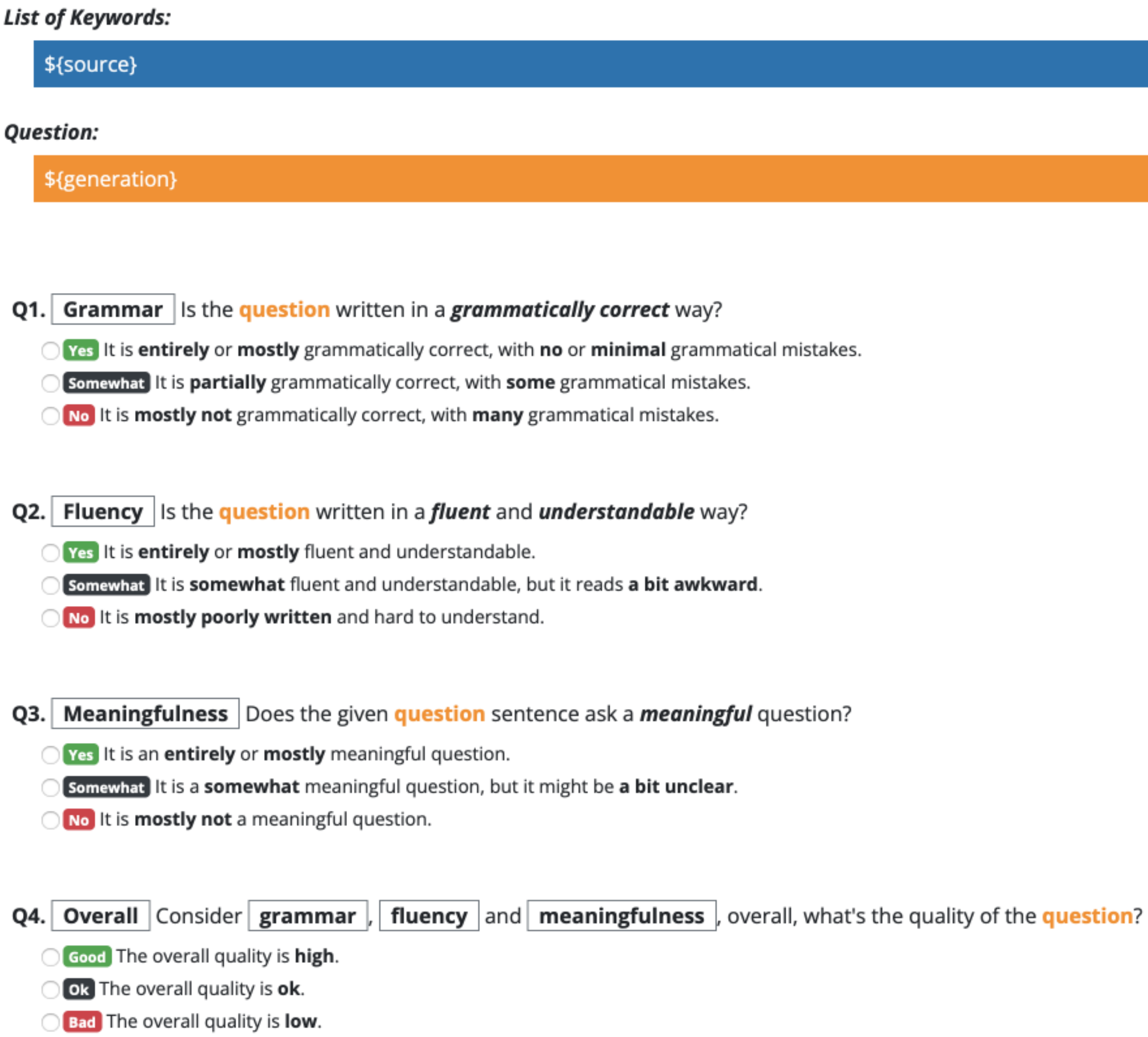
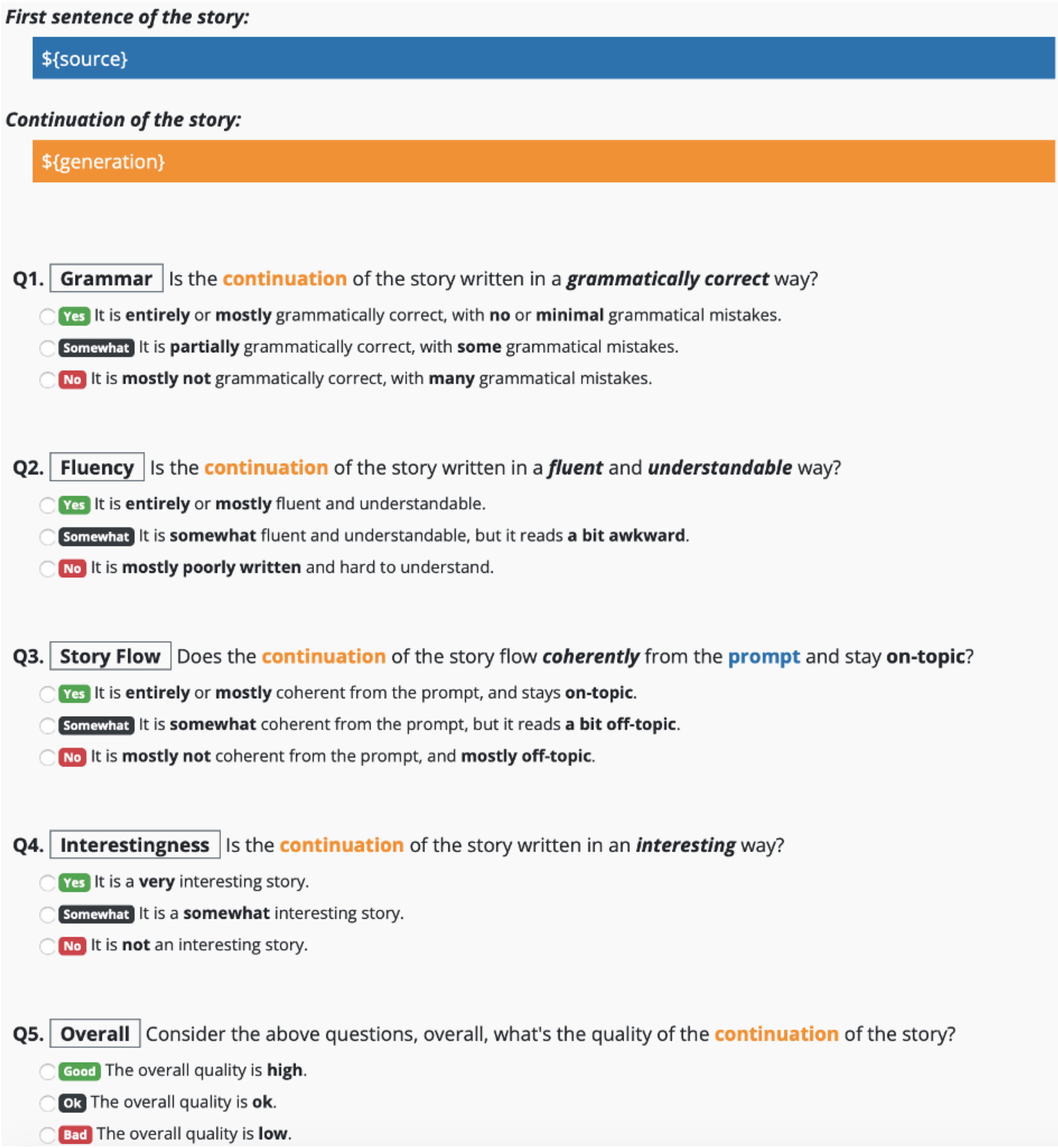


Figure 6: Human evaluation template for the Interrogative Sentence Generation task.


First sentence of the story:

${source}

Continuation of the story:

${generation}

Q1. **Grammar** Is the **continuation** of the story written in a ***grammatically correct*** way?

- **Yes** It is **entirely** or **mostly** grammatically correct, with **no** or **minimal** grammatical mistakes.
- **Somewhat** It is **partially** grammatically correct, with **some** grammatical mistakes.
- **No** It is **mostly not** grammatically correct, with **many** grammatical mistakes.

Q2. **Fluency** Is the **continuation** of the story written in a ***fluent*** and ***understandable*** way?

- **Yes** It is **entirely** or **mostly** fluent and understandable.
- **Somewhat** It is **somewhat** fluent and understandable, but it reads **a bit awkward**.
- **No** It is **mostly poorly written** and hard to understand.

Q3. **Story Flow** Does the **continuation** of the story flow ***coherently*** from the **prompt** and stay **on-topic**?

- **Yes** It is **entirely** or **mostly** coherent from the prompt, and stays **on-topic**.
- **Somewhat** It is **somewhat** coherent from the prompt, but it reads **a bit off-topic**.
- **No** It is **mostly not** coherent from the prompt, and **mostly off-topic**.

Q4. **Interestingness** Is the **continuation** of the story written in an ***interesting*** way?

- **Yes** It is a **very** interesting story.
- **Somewhat** It is a **somewhat** interesting story.
- **No** It is **not** an interesting story.

Q5. **Overall** Consider the above questions, overall, what's the quality of the **continuation** of the story?

- **Good** The overall quality is **high**.
- **Ok** The overall quality is **ok**.
- **Bad** The overall quality is **low**.

Figure 7: Human evaluation template for the RocStories task.