SAPPHIRE: Approaches for Enhanced Concept-to-Text Generation

Steven Y. Feng, Jessica Huynh, Chaitanya Narisetty, Eduard Hovy, Varun Gangal
Language Technologies Institute
Carnegie Mellon University
{syfeng,jhuynh,cnariset,hovy,vgangal}@cs.cmu.edu

Abstract

We motivate and propose a suite of simple but effective improvements for concept-to-text generation called SAPPHIRE: Set Augmentation and Post-hoc PHrase Infilling and REcombination. We demonstrate their effectiveness on generative commonsense reasoning, a.k.a. the *CommonGen* task, through experiments using both BART and T5 models. Through extensive automatic and human evaluation, we show that SAPPHIRE noticeably improves model performance. An in-depth qualitative analysis illustrates that SAPPHIRE effectively addresses many issues of the baseline model generations, including lack of commonsense, insufficient specificity, and poor fluency.

1 Introduction

There has been increasing interest in constrained text generation tasks which involve constructing natural language outputs under certain preconditions, such as particular words that must appear in the output sentences. A related area of work is data-to-text natural language generation (NLG), which requires generating natural language descriptions of structured or semi-structured data inputs. Many constrained text generation and NLG tasks share commonalities, one of which is their task formulation: a set of inputs must be converted into natural language sentences. This set of inputs can be, in many cases, thought of as *concepts*, e.g. higher-level words or structures that play an important role in the generated text.

With the increased popularity of Transformer-based models and their application to many NLP tasks, performance on many text generation tasks has improved considerably. Much progress in recent years has been from the investigation of model improvements, such as larger and more effectively pretrained language generation models. However, are there simple and effective approaches to improving performance on these tasks that can come from the data itself? Further, can we potentially use the outputs of these models themselves to further improve their task performance - a "self-introspection" of sorts?

In this paper, we show that the answer is yes. We propose a suite of simple but effective improvements for concept-to-text generation called SAPPHIRE: Set Augmentation and Post-hoc PHrase Infilling and REcombination. Specifically, SAPPHIRE is composed of two major approaches: 1) the augmentation of input concept sets (§4.1), 2) the recombination of phrases extracted from baseline generations into more fluent and logical text (§4.2). These are mainly model-agnostic improvements that rely on the data itself and the model's own initial generations, respectively.[1]

We focus on generative commonsense reasoning, or CommonGen (Lin et al., 2020), which involves generating logical sentences describing an everyday scenario from a set of concepts, which in this case are individual words that must be represented in the output text in some form. CommonGen is a challenging instance of constrained text generation that assesses 1) relational reasoning abilities using commonsense knowledge, and 2) compositional generalization capabilities to piece together concept combinations. Further, CommonGen's task formulation and evaluation methodology are quite broadly applicable and encompassing, making it a good benchmark for general constrained text generation capability. Further, this is an opportune moment to investigate this task as commonsense ability of NLP models, particularly for generation, has received increasing community attention through works like COMET (Bosselut et al., 2019).

We perform experiments on varying sizes of two

[1] Code at https://github.com/styfeng/SAPPHIRE

Proceedings of the 14th International Conference on Natural Language Generation (INLG), pages 212–225,
Aberdeen, Scotland, UK, 20-24 September 2021. ©2021 Association for Computational Linguistics

Dataset Stats	Train$_{CG}$	Dev$_O$	Test$_O$	Dev$_{CG}$	Test$_{CG}$
# concept sets	32,651	993	1,497	240	360
size = 3	25,020	493	-	120	-
size = 4	4,240	250	747	60	180
size = 5	3,391	250	750	60	180
# sentences	67,389	4,018	7,644	984	1583

Table 1: CommonGen dataset statistics.

state-of-the-art Transformer-based language generation models: BART (Lewis et al., 2020) and T5 (Raffel et al., 2020). We first conduct an extensive correlation study (§3.1) and qualitative analysis (§3.2) of these models' generations after simply training on CommonGen. We find that performance is positively correlated with concept set size, motivating concept set augmentation. We also find that generations contain issues related to commonsense and fluency which can possibly be addressed through piecing the texts back together in different ways, motivating phrase recombination.

Fleshing out our first intuition - we devise two methods to augment concepts from references during training through extracted keywords (§4.1.1) and attention matrices (§4.1.2). For the phrase recombination intuition, we propose two realizations based on a new training stage (§4.2.1) and masked infilling (§4.2.2). Finally, through comprehensive evaluation (§6), we show how the SAPPHIRE suite drives up model performance across metrics, besides addressing aforementioned baseline deficiencies on commonsense, specificity, and fluency.

2 Dataset, Models, and Metrics

2.1 CommonGen Dataset

The CommonGen dataset is split into train, dev, and test splits, covering a total of 35,141 concept sets and 79,051 sentences. The concept sets range from 3 keywords to 5 keywords long. As the original test set is hidden, we split the provided dev set into a new dev and test split for the majority of our experiments while keeping the training split untouched. Note that we also evaluate our SAPPHIRE models on the original test set with help from the CommonGen authors (see §6.1). We will henceforth refer to these new splits as train$_{CG}$, dev$_{CG}$, and test$_{CG}$, and the original dev and test splits as dev$_O$ and test$_O$. The statistics of our new splits compared to the originals can be found in Table 1. We attempt to keep the relative sizes of the new dev and test splits and the distribution of concept set sizes within each split similar to the originals.

Model\Metrics	BLEU-4	CIDEr	SPICE
Reported BART-large	27.50	14.12	30.00
Reported T5-base	18.00	9.73	23.40
Reported T5-Large	30.60	15.84	31.80
Our BART-base	28.30	15.07	30.35
Our BART-large	**30.20**	**15.72**	**31.20**
Our T5-base	**31.00**	**16.37**	**32.05**
Our T5-large	**33.60**	**17.02**	**33.45**

Table 2: Performance of our re-implemented CommonGen models on dev$_O$ compared to a subset of original numbers reported in Lin et al. (2020). For our models, results are averaged over two seeds. The original authors did not experiment with BART-base. Bold indicates where we match or exceed the reported metric. See §2.3 for explanations of the metrics and Appendix B for a full metric comparison table.

2.2 Models: T5 and BART

We perform experiments using pretrained language generators, specifically BART and T5 (both base and large versions). BART (Lewis et al., 2020) is a Transformer-based seq2seq model trained as a denoising autoencoder to reconstruct original text from noised text. T5 (Raffel et al., 2020) is another seq2seq Transformer with strong multitask pretraining. We use their HuggingFace codebases.

We train two seeded instances of each model on train$_{CG}$, evaluating their performance on dev$_O$, and comparing our numbers to those reported in Lin et al. (2020) to benchmark our implementations. These essentially serve as the four baseline models for our ensuing experiments. We follow the hyperparameters from Lin et al. (2020), choose the epoch reaching highest ROUGE-2 on the dev split, and use beam search for decoding.[2] From Table 2, we see that our re-implemented models match or exceed the original reported results on most metrics across different models.

2.3 Evaluation Metrics

For our experiments, we use a gamut of automatic evaluation metrics. These include those used by Lin et al. (2020), such as BLEU (Papineni et al., 2002), CIDEr (Vedantam et al., 2015), SPICE (Anderson et al., 2016), and Coverage (Cov). Barring Cov, these metrics measure the similarity between generated text and human references. Cov measures the average % of input concepts covered by the generated text. We also introduce BERTScore (Zhang et al., 2020), which measures token-by-token BERT (Devlin et al., 2019) embeddings similarity. It also measures the similarity between the generated text and human references, but on a more semantic (rather than surface token) level. When

[2] See Appendix A for further details.

reporting BERTScore, we multiply by 100. For all metrics, higher corresponds to better performance.

3 Initial Analysis

3.1 Correlation Study

We begin by conducting an analysis of the four baselines implemented and discussed in §2.2, which we refer to henceforth as BART-base-BL, BART-large-BL, T5-base-BL, and T5-large-BL. One aspect we were interested in is whether the number of input concepts affects the quality of generated text. We conduct a comprehensive correlation study of the performance of the four baselines on dev$_O$ w.r.t. the number of input concepts.

As seen from Table 3, the majority of the metrics are positively correlated with concept set size across the models. ROUGE-L, CIDEr, and SPICE have small correlations that are mainly statistically insignificant, demonstrating that they are likely uncorrelated with concept set size. Coverage is strongly negatively correlated, showing that there is a higher probability of concepts missing from the generated text as concept set size increases.

There are two major takeaways from this. Firstly, increased concept set size results in greater overall performance. Secondly, models have difficulty with coverage given increased concept set size. This motivates our first set of improvements, which involves augmenting the concept sets with additional words in hopes of 1) increasing performance of the models and 2) improving their coverage, as we hope that training with more input concepts will help models learn to better cover them in the generated text. This is discussed more in §4.1.

3.2 Qualitative Analysis

We conduct a qualitative analysis of the baseline model outputs. We observe that several outputs are more like phrases than full coherent sentences, e.g. *"body of water on a raft"*. Some generated texts are also missing important words, e.g. *"A listening music and dancing in a dark room"* is clearly missing a noun before *listening*. A large portion of generated texts are quite generic and bland, e.g. *"Someone sits and listens to someone talk"*, while more detailed and specific statements are present in the human references. This can be seen as an instance of the noted *"dull response" problem* faced by generation models (Du and Black, 2019; Li et al., 2016), where they prefer safe, short, and frequent responses independent of the input.

Another issue is the way sentences are pieced together. Certain phrases in the outputs are either joined in the wrong order or with incorrect connectors, leading to sentences that appear to lack commonsense. For example, *"body of water on a raft"* is illogical, and the phrases *"body of water"* and *"a raft"* are pieced together incorrectly. Example corrections include *"body of water carrying a raft"* and *"a raft on a body of water"*. The first changes the adverb *on* joining them to the verb *carrying*, and the second pieces them together in the opposite order. A similar issue occurs with the {*horse, carriage, draw*} example in Table 4.

Some major takeaways are that many generations are: 1) phrases rather than full sentences and 2) poorly pieced together and lack fluency and logic compared to human references. This motivates our second set of improvements, which involves recombining extracted phrases from baseline generations into hopefully more fluent and logical sentences. This is discussed more in §4.2.

4 SAPPHIRE Methodology

4.1 Concept Set Augmentation

The first set of improvements is concept set augmentation, which involves augmenting the input concept sets. We try augmentation using up to 1 to 5 additional words, and train-time augmentation both with and without test-time augmentation. We observed that test-time augmentation resulted in inconsistent results that were not as effective, and stick with train-time only augmentation. During training, rather than feeding in the original concept sets as inputs, we instead feed in these augmented concept sets which consist of more words. The expected outputs are the same human references. During test-time, we simply feed in the original concept sets (without augmentation) as inputs.

4.1.1 Keyword-based Augmentation

The first type of augmentation we try is keyword-based, or *Kw-aug*. We augment the train$_{CG}$ concept sets with keywords extracted from the human references using KeyBERT[3] (Grootendorst, 2020). We calculate the average semantic similarity (using cosine similarity of BERT embeddings) of the candidate keywords to the original concept set. At each stage of augmentation, we add the remaining can-

[3] https://github.com/MaartenGr/KeyBERT

	BART-base			BART-large			T5-base			T5-large		
Correlation	PCC	ρ	τ	PCC	ρ	τ	PCC	ρ	τ	PCC	ρ	τ
ROUGE-1	0.08	0.09	0.07	0.10	0.12	0.09	0.04	0.05	0.04	0.10	0.11	0.09
ROUGE-2	0.05	0.08	0.07	0.05	0.10	0.07	0.03	0.07	0.05	0.06	0.09	0.07
ROUGE-L	0.00*	0.01*	0.01*	0.00*	0.02*	0.01*	-0.03	-0.01*	-0.01*	0.02*	0.04	0.03
BLEU-1	0.08	0.08	0.06	0.14	0.14	0.11	0.00*	0.03*	0.02*	0.08	0.11	0.09
BLEU-2	0.06	0.06	0.04	0.11	0.11	0.08	0.03*	0.04*	0.03*	0.09	0.10	0.07
BLEU-3	0.08	0.06	0.05	0.09	0.09	0.06	0.04*	0.03*	0.02*	0.09	0.08	0.06
BLEU-4	0.05	0.05	0.04	0.05	0.07	0.05	0.04*	0.02*	0.02*	0.08	0.08	0.06
METEOR	0.05	0.08	0.06	0.06	0.09	0.07	0.02*	0.04	0.03	0.06	0.08	0.06
CIDEr	-0.02*	-0.03*	-0.02*	0.01*	0.02*	0.02*	-0.08	-0.10	-0.07	0.00*	0.00*	0.00*
SPICE	-0.02*	-0.01*	-0.01*	0.01*	0.02*	0.01*	-0.02*	-0.02*	-0.02*	0.02*	0.03*	0.02*
BERTScore	0.04	0.03	0.02	0.06	0.06	0.05	0.04	0.03	0.02	0.05	0.04	0.03
Coverage	-0.26	-0.31	-0.27	-0.07	-0.13	-0.11	-0.38	-0.42	-0.37	-0.26	-0.31	-0.28

Table 3: Correlations on dev$_O$ between concept set size and evaluation metrics for our four baseline models (over the results from both seeds); values marked with * are statistically insignificant. PCC refers to Pearson correlation coefficient, ρ to Spearman's rank correlation coefficient, and τ to Kendall rank correlation coefficient.

Concept Set	Baseline Generation	Human Reference
{horse, carriage, draw}	horse drawn in a carriage	The carriage is drawn by the horse.
{fish, catch, pole}	fish caught on a pole	The man used a fishing pole to catch fish.
{listen, talk, sit}	Someone sits and listens to someone talk.	The man told the boy to sit down and listen to him talk.
{bathtub, bath, dog, give}	A dog giving a bath in a bathtub.	The teenager made a big mess in the bathtub giving her dog a bath.

Table 4: Example generations from our baseline models versus human references.

Method	Original Concept Set	Added Words
Kw-aug	{match, stadium, watch}	{soccer, league, fans}
Kw-aug	{family, time, spend}	{holidays}
Kw-aug	{head, skier, slope}	{cabin}
Att-aug	{boat, lake, drive}	{fisherman}
Att-aug	{family, time, spend}	{at, holidays}
Att-aug	{player, match, look}	{tennis, on, during}

Table 5: Example train$_{CG}$ concept set augmentations.

didate with the highest similarity.[4] Some augmentation examples can be found in Table 5. We train our BART and T5 models using these augmented sets, and call the resulting models BART-base-KW, BART-large-KW, T5-base-KW, and T5-large-KW.

4.1.2 Attention-based Augmentation

We also try attention-based augmentation, or *Att-aug*. We augment the train$_{CG}$ concept sets with the words that have been most attended upon in aggregate by the other words in the human references. We pass the reference texts through BERT and return the attention weights at the last layer. At each stage of augmentation, we add the remaining candidate word with the highest attention. Adding the least attended words would not be effective as many are words with little meaning (e.g. simple articles such as *“a”* and “the”). Some augmentation examples can be found in Table 5. We train our BART and T5 models using these augmented sets, and call the resulting models BART-base-Att, BART-large-Att, T5-base-Att, and T5-large-Att.

4.2 Phrase Recombination

The second set of improvements is phrase recombination, which involves breaking down sentences into phrases and reconstructing them into new sentences which are hopefully more logical and coherent. For training, we use YAKE (Campos et al., 2018) to break down the train$_{CG}$ human references into phrases of up to 2, 3, and 5 n-grams long, and ensure extracted phrases have as little overlap as possible. During validation and testing, since we assume no access to ground-truth human references, we instead use YAKE to extract keyphrases from our baseline model generations.

We ignore extracted 1-grams as this approach focuses on phrases. We find words from the original concept set which are not covered by our extracted keyphrases and include them to ensure that coverage is maintained. Essentially, we form a new concept set which can also consist of phrases. Some examples can be found in Table 6.

4.2.1 Phrase-to-text (P2T)

To piece the phrases back together, we try phrase-to-text (P2T) generation by training BART and T5 to generate full sentences given our new input sets, and call these models BART-base-P2T, BART-large-P2T, T5-base-P2T, and T5-large-P2T. During

[4] We also tried using the least semantically similar keywords, but results were noticeably worse.

Original Text	Extracted Keyphrases	New Input Concept Set
A dog wags his tail at the boy.	dog wags his tail	{dog wags his tail}
hanging a painting on a wall at home	hanging a painting	{hanging a painting, wall}
a herd of many sheep crowded together in a stable waiting to be dipped for ticks and other pests	herd of many sheep crowded	{herd of many sheep crowded, dip, waiting}
a soldier takes a knee while providing security during a patrol outside of the village.	knee while providing security, patrol outside of the village	{knee while providing security, patrol outside of the village, take}

Table 6: Example keyphrases (up to 5-grams) extracted using YAKE from human-written training references.

training, we choose a single random permutation of each training input set (consisting of extracted keyphrases from the human references), with the elements separated by <*s*>, and the human references as the outputs. This is in order for the models to learn to be order-agnostic, which is important as one desired property of phrase recombination is the ability to combine phrases in different orders, as motivated by the qualitative analysis in §3.2. During inference or test-time, we feed in a single random permutation of each test-time input set, consisting of extracted keyphrases from the corresponding baseline model's outputs.

4.2.2 Mask Infilling (MI)

This method interpolates text between test-time input set elements with no training required. For example, given a test-time input set $\{c_1, c_2\}$, we feed in "*<mask>* c_1 *<mask>* c_2 *<mask>*" and "*<mask>* c_2 *<mask>* c_1 *<mask>*" to an MI model to fill the <mask> tokens with text. We use BART-base and BART-large for MI, and call the approaches BART-base-MI and BART-large-MI, respectively. We use BART-base-MI on input sets consisting of extracted keyphrases from BART-base-BL and T5-base-BL, and BART-large-MI on input sets consisting of extracted keyphrases from BART-large-BL and T5-large-BL. We also try MI on the original concept sets (with no phrases).

One difficulty is determining the right input set permutation. Many contain ≥5 elements (meaning ≥5!=120 permutations), making exhaustive MI infeasible. Order of elements for infilling can result in vastly different outputs (see §6.3), as certain orders are more natural. Humans perform their own intuitive reordering of given inputs when writing, and the baselines and other approaches (e.g. Kw-aug, P2T) learn to mainly be order agnostic.

We use perplexity (PPL) from GPT-2 (Radford et al., 2019) to pick the *"best"* permutations for MI. We feed up to 120 permutations of each input set (with elements separated by spaces) to GPT-2 to extract their PPL, and keep the 10 with lowest PPL per example. This is not a perfect approach, but is likely better than random sampling. For each example, we perform MI on these ten permutations, and select the output with lowest GPT-2 PPL.

We found BART-large-MI outputs contain URLs, news agency names in brackets, etc. Hence, we post-process before output selection and evaluation. BART-base-MI does not do this. One possible explanation is that BART-large may have been pretrained on more social media and news data.

5 Experiments

5.1 Model Training and Selection

For training Kw-aug, Att-aug, and P2T models, we follow baseline hyperparameters, barring learning rate (LR) which is tuned per-method. We train two seeds per model. See Appendix A for more.

For each model, we choose the epoch corresponding to highest ROUGE-2 on the dev split, and use beam search for decoding. The dev and test splits are different. For Kw-aug and Att-aug models, the splits are simply dev$_{CG}$ and test$_{CG}$ (or test$_{O}$), as we do not perform test-time augmentation. For P2T, the splits are dev$_{CG}$ and test$_{CG}$ (or test$_{O}$) but with the input sets replaced with new ones that include keyphrases extracted from the corresponding baseline model's outputs.

The number of words to augment for Kw-aug and Att-aug (from 1 to 5) and maximum n-gram length of extracted keyphrases for P2T (2, 3, or 5) are hyperparameters. While we train separate versions of each model corresponding to different values of these, the final chosen model per method and model combination (such as BART-base-KW) is the one corresponding to the hyperparameter value that performs best on the dev split when averaged over both seeds. For MI, which involves no training, we select the variation (MI on the original concept set or new input sets with keyphrases up to 2, 3, or 5 n-grams) per model that performs best on the dev split, and only perform infilling using extracted keyphrases from the first seed baseline generations. These are the selected models we report the test$_{CG}$ and test$_{O}$ results of in §6.

5.2 Human Evaluation

We ask annotators to evaluate 48 test$_{CG}$ examples from the human references, baseline outputs, and various method (excluding MI) outputs for BART-large and T5-base. We choose these two as they cover both model types and sizes, and exclude MI as it performs noticeably worse on the automatic evaluation (see §6.1). See Appendix §C for more.

The annotators evaluate fluency and commonsense of the texts on 1-5 scales. Fluency, also known as naturalness, is a measure of how human-like a text is. Commonsense is the plausibility of the events described. We do not evaluate coverage as automatic metrics suffice; coverage is more objective compared to fluency and commonsense.

6 Results and Analysis

Automatic evaluation results on test$_{CG}$ can be found in Tables 7, 8, 9, 10, and results on test$_{O}$ in Table 12. Human evaluation results on test$_{CG}$ can be found in Table 13. Single keyword augmentation performs best for Kw-aug across models. Two word augmentation performs best for Att-aug, except T5-base where three word augmentation performs best. Keyphrases up to 2-grams long perform best for P2T, except T5-large where 3-grams perform best. All models perform best with keyphrases up to 5-grams long for MI. These are the results reported here, and graphs displaying other hyperparameter results on test$_{CG}$ are in Appendix D. Table 14 contains qualitative examples, and more can be found in Appendix §E.

6.1 Automatic Evaluation

We see from Tables 7 to 10 that SAPPHIRE methods outperform the baselines on most/all metrics across the models on test$_{CG}$. The only exception is MI, which performs worse other than coverage.

For BART-base, Kw-aug, Att-aug, and P2T all outperform the baseline across the metrics. For BART-large, Att-aug and P2T outperform the baseline heaviest, with noticeable increases to all metrics. For T5-base, all methods outperform the baseline, with Kw-aug performing best. Att-aug performs best for T5-large, and SAPPHIRE appears relatively less effective for T5-large. T5-large is the strongest baseline, and hence further improving its performance is possibly more difficult.

MI performs worse across most metrics except coverage, likely as MI almost always keeps inputs intact in their exact form. This is however possibly one reason for its low performance; it is less flexible. Further, as discussed in §4.2.2, MI is highly dependent on the input order. See §6.3 for more.

Table 11 contains statistical significance p-values from Pitman's permutation tests (Pitman, 1937) for what we adjudged to be the best performing method(s) per model compared to corresponding baselines on test$_{CG}$. Most metrics across the methods are significant compared to the baselines.

From Table 12, we see that SAPPHIRE models outperform the corresponding baselines reported in Lin et al. (2020) on test$_{O}$. T5-large-KW and P2T outperform EKI-BART (Fan et al., 2020) and KG-BART (Liu et al., 2021) on both SPICE and BLEU-4, which are two SOTA published CommonGen models that use external knowledge from corpora and KGs. As SPICE is used to rank the CommonGen leaderboard[5], T5-large-KW and P2T would place highly. SAPPHIRE models do lag behind the SOTA published RE-T5 (Wang et al., 2021), showing potential for further improvement. Further, the BART-large SAPPHIRE models perform worse than EKI-BART and KG-BART, but not by a substantial margin. We emphasize again that SAPPHIRE simply uses the data itself and the baseline generations, rather than external knowledge. Hence, SAPPHIRE's performance gains over the baselines and certain SAPPHIRE models matching or outperforming SOTA models that leverage external information is quite impressive.

6.2 Human Evaluation

Table 13 shows human evaluation results on test$_{CG}$ for human references and methods (excluding MI) using BART-large and T5-base. SAPPHIRE generally outperforms the baselines. BART-large-P2T performs noticeably higher on both fluency and commonsense. For T5-base, all three methods outperform the baseline across both metrics. Compared to humans, our best methods have comparable fluency, but still lag noticeably on commonsense, demonstrating that human-level generative commonsense reasoning is indeed challenging.

6.3 Qualitative Analysis

We see from Table 14 that many baseline outputs contain issues found in §3.2, e.g. incomplete or illogical sentences. Human references are fluent, logical, and sometimes more creative (e.g. example 5), which all methods still lack in comparison.

[5] https://inklab.usc.edu/CommonGen/leaderboard.html

	BART-base				
Metrics\Methods	Baseline	Kw-aug	Att-aug	P2T	BART-base-MI
ROUGE-1	43.96±0.03	**45.01**±0.00	44.99±0.10	44.87±0.42	44.83
ROUGE-2	17.31±0.02	**18.33**±0.06	18.18±0.04	18.04±0.13	17.44
ROUGE-L	36.65±0.00	37.28±0.24	**37.76**±0.12	37.28±0.11	34.47
BLEU-1	**73.20**±0.28	73.00±0.85	73.00±0.14	73.15±1.06	69.90
BLEU-2	54.50±0.14	55.35±0.49	**55.70**±0.28	55.65±0.35	49.00
BLEU-3	40.40±0.14	41.35±0.21	41.40±0.28	**41.85**±0.35	34.70
BLEU-4	30.10±0.14	31.10±0.14	30.95±0.07	**31.75**±0.35	24.70
METEOR	30.35±0.35	30.50±0.28	30.70±0.14	**31.05**±0.49	29.70
CIDEr	15.56±0.10	**16.18**±0.12	15.68±0.00	16.14±0.33	14.43
SPICE	30.05±0.07	30.45±0.07	30.65±0.35	**30.95**±0.21	28.40
BERTScore	59.19±0.32	59.32±0.25	**59.72**±0.03	59.54±0.05	53.73
Coverage	90.43±0.17	91.44±0.95	91.23±0.21	91.47±2.93	**96.23**

Table 7: Automatic evaluation results (with standard deviations) for BART-base on test$_{CG}$, averaged over two seeds for trained models. Bold corresponds to best performance on that metric.

	BART-large				
Metrics\Methods	Baseline	Kw-aug	Att-aug	P2T	BART-large-MI
ROUGE-1	45.67±0.25	46.71±0.05	**46.78**±0.14	46.26±0.29	41.69
ROUGE-2	18.77±0.04	19.64±0.05	**19.92**±0.19	19.37±0.17	15.40
ROUGE-L	37.83±0.29	38.38±0.01	**38.53**±0.03	38.22±0.16	33.32
BLEU-1	74.45±0.21	76.20±0.99	76.55±0.92	**77.10**±0.85	63.90
BLEU-2	56.25±0.78	58.60±0.14	**59.60**±0.00	58.95±0.64	42.40
BLEU-3	42.15±0.49	44.00±0.00	**45.20**±0.42	44.70±0.14	29.20
BLEU-4	32.10±0.42	33.40±0.28	**34.50**±0.42	34.25±0.21	20.50
METEOR	31.70±0.14	32.60±0.57	32.65±0.49	**33.00**±0.14	28.30
CIDEr	16.42±0.09	17.37±0.08	17.49±0.49	**17.50**±0.02	12.32
SPICE	31.85±0.21	33.15±0.49	33.30±0.28	**33.60**±0.00	26.10
BERTScore	59.95±0.29	60.83±0.29	60.87±0.45	**61.30**±0.66	48.56
Coverage	94.49±0.53	96.74±1.20	96.02±1.17	**97.02**±0.15	95.33

Table 8: Automatic evaluation results (with standard deviations) for BART-large on test$_{CG}$, averaged over two seeds for trained models. Bold corresponds to best performance on that metric.

	T5-base				
Metrics\Methods	Baseline	Kw-aug	Att-aug	P2T	BART-base-MI
ROUGE-1	44.63±0.13	46.42±0.01	**46.75**±0.11	45.73±0.27	44.92
ROUGE-2	18.40±0.14	**19.36**±0.24	19.20±0.17	18.51±0.11	17.98
ROUGE-L	37.60±0.16	**38.68**±0.08	38.51±0.21	38.07±0.10	34.88
BLEU-1	73.60±0.85	**76.25**±0.35	76.00±0.28	75.65±1.20	70.20
BLEU-2	57.00±0.71	**59.55**±0.64	58.75±0.35	58.15±0.64	50.50
BLEU-3	42.75±0.49	**45.10**±0.85	44.00±0.28	43.45±0.07	36.20
BLEU-4	32.70±0.42	**34.45**±0.92	33.30±0.28	33.10±0.28	26.10
METEOR	31.05±0.49	31.85±0.07	31.90±0.14	**32.05**±0.35	30.20
CIDEr	16.26±0.25	**17.37**±0.04	17.04±0.21	16.84±0.11	14.83
SPICE	31.95±0.07	32.75±0.21	32.85±0.21	**33.20**±0.14	29.70
BERTScore	61.40±0.34	**61.88**±0.06	61.28±0.10	61.46±0.01	55.04
Coverage	90.96±1.77	94.92±0.45	96.00±0.03	94.78±0.83	**96.03**

Table 9: Automatic evaluation results (with standard deviations) for T5-base on test$_{CG}$, averaged over two seeds for trained models. Bold corresponds to best performance on that metric.

	T5-large				
Metrics\Methods	Baseline	Kw-aug	Att-aug	P2T	BART-large-MI
ROUGE-1	46.26±0.17	**47.47**±0.16	47.40±0.12	46.72±0.26	42.78
ROUGE-2	19.62±0.17	20.02±0.07	**20.19**±0.01	19.76±0.22	16.61
ROUGE-L	39.21±0.22	39.84±0.12	**39.97**±0.06	39.19±0.09	34.52
BLEU-1	77.45±0.21	78.70±0.28	**78.95**±0.07	77.90±0.57	66.80
BLEU-2	60.75±0.21	62.10±0.14	**62.35**±0.07	61.00±0.42	45.90
BLEU-3	46.60±0.14	47.65±0.21	**47.95**±0.21	46.75±0.49	32.70
BLEU-4	36.30±0.00	36.80±0.28	**37.35**±0.49	36.10±0.42	23.90
METEOR	33.30±0.14	33.55±0.07	**33.70**±0.00	33.35±0.21	29.10
CIDEr	17.90±0.15	18.40±0.18	**18.43**±0.10	17.89±0.08	13.34
SPICE	34.25±0.07	**34.50**±0.28	33.70±0.14	34.00±0.28	28.00
BERTScore	62.65±0.07	**62.91**±0.15	62.78±0.21	62.46±0.11	50.57
Coverage	94.23±0.21	95.92±0.02	**96.08**±0.09	95.44±0.58	96.03

Table 10: Automatic evaluation results (with standard deviations) for T5-large on test$_{CG}$, averaged over two seeds for trained models. Bold corresponds to best performance on that metric.

	BART-base	**BART-large**		**T5-base**	**T5-large**
p-values	P2T	Att-aug	P2T	Kw-aug	Att-aug
ROUGE-1	1.58E-05	1.58E-05	7.58E-04	1.58E-05	1.58E-05
ROUGE-2	6.32E-05	1.58E-05	2.18E-03	1.58E-05	2.20E-03
ROUGE-L	6.32E-05	8.53E-04	2.78E-02	1.58E-05	1.58E-05
BLEU-1	**3.63E-01**	1.39E-04	6.94E-05	6.94E-05	1.11E-03
BLEU-2	1.11E-03	6.94E-05	6.94E-05	6.94E-05	5.69E-03
BLEU-3	3.26E-02	1.04E-03	9.03E-04	4.17E-04	3.40E-02
BLEU-4	**5.68E-02**	**1.57E-01**	8.40E-03	1.83E-02	**2.66E-01**
METEOR	1.57E-02	9.03E-04	6.94E-05	2.08E-04	**7.27E-01**
CIDEr	6.25E-04	2.08E-04	6.94E-05	6.94E-05	5.07E-03
SPICE	1.53E-03	6.25E-04	6.94E-05	1.43E-02	**9.16E-01**
BERTScore	3.33E-03	1.58E-05	1.58E-05	1.58E-05	**1.42E-01**
Coverage	3.16E-05	1.58E-05	1.58E-05	1.58E-05	1.58E-05

Table 11: Statistical significance p-values (from Pitman's permutation tests) for the best performing method(s) per model compared to the corresponding baselines. Insignificant p-values (using $\alpha = 0.05$ or 5E-02) are bolded.

For example 1, the baseline generation *"hands sitting on a chair"* misses the concept *"toy"*, whereas our methods do not. Kw-aug and Att-aug output complete and logical sentences. For example 2, the baseline generation of *"a camel rides a camel"* is illogical. Our methods output more logical and specific sentences. For example 3, our methods generate more complete and coherent sentences than the baseline, which lacks a subject (does not mention *who* is *"walking"*). For example 4, the baseline generation *"bus sits on the tracks"* is illogical as buses park on roads. Our methods do not suffer from this and output more reasonable text. For example 5, the baseline generation *"A lady sits in a sunglass."* is completely illogical. Kw-aug, Att-aug, and P2T all output logical text. For example 6, the baseline output *"Someone stands in front of someone holding a hand"* is generic and bland. Kw-aug, Att-aug, and P2T all output more specific and detailed text rather than simply referring to *"someone"*. Overall, SAPPHIRE generates text that is more complete, fluent, logical, and with greater coverage, addressing many baseline issues (§3.2).

However, SAPPHIRE methods are imperfect. P2T relies heavily on the original generation. For example 1, the baseline output *"hands sitting on a chair"* is extracted as a keyphrase, and used in the P2T output *"hands sitting on a chair with toys"*. While coverage improves, the text is still illogical. For example 2, P2T still misses the *"walk"* concept. While the Att-aug output of *"A man is riding camel as he walks through the desert."* is more logical than the baseline's, it is still not entirely logical as the man cannot ride the camel and walk at the same time. MI outputs logical and fluent text for examples 2 and 3. For the other examples, the generated texts are illogical, not fluent, or incomplete.

This is likely due to input permutation having a strong effect on output quality. For example, *"wave"* before *"falls off a surf board"* leads to an illogical output *"A wave falls off a surf board."*, where the reverse order results in a more logical output *"A man falls off a surf board and hits a wave."* As discussed in §4.2.2, our method of selecting best permutations is likely imperfect. Further, BART-MI usually does not inflect inputs and retains them in exact form, unlike the baselines and other methods which learn to inflect words (e.g. singular to plural). We believe BART-MI has potential if these weaknesses can be addressed.

7 Related Work

Constrained Text Generation: There has been more work on constrained text generation. Miao et al. (2019) use Metropolis-Hastings sampling to determine token-level edits at each step of generation. Feng et al. (2019) introduce Semantic Text Exchange to adjust the semantics of a text given a *replacement entity*. Gangal et al. (2021a) propose narrative reordering (NAREOR) to rewrite stories in different narrative orders while preserving plot.

Data-to-text NLG: A wide range of data-to-text NLG benchmarks have been proposed, e.g. for generating weather reports (Liang et al., 2009), game commentary (Jhamtani et al., 2018), and recipes (Kiddon et al., 2016). E2E-NLG (Dušek et al., 2018) and WebNLG (Gardent et al., 2017) are two benchmarks that involve generating text from meaning representation (MR) and triple sequences. Montella et al. (2020) use target Wiki sentences with parsed OpenIE triples as weak supervision for WebNLG. Tandon et al. (2018) permute input MRs to augment examples for E2E-NLG.

Commonsense Reasoning and Incorporation: Talmor et al. (2020) show that not all pretrained LMs can reason through commonsense tasks. Other works investigate commonsense injection into models; one popular way is through knowledge graphs (KGs). One large commonsense KG is COMET, which trains on KG edges to learn connections between words and phrases. COSMIC (Ghosal et al., 2020) uses COMET to inject commonsense. EKI-BART (Fan et al., 2020) and KG-BART (Liu et al., 2021) show that external knowledge (from corpora and KGs) can improve performance on CommonGen. Distinctly, SAPPHIRE obviates reliance on external knowledge.

Models\Metrics	ROUGE-2/L		**BLEU-3/4**		METEOR	**CIDEr**	**SPICE**	Coverage
T5-base (reported baseline)	14.63	34.56	28.76	18.54	23.94	9.40	19.87	76.67
BART-large (reported baseline)	22.02	41.78	39.52	29.01	31.83	13.98	28.00	97.35
T5-large (reported baseline)	21.74	42.75	43.01	31.96	31.12	15.13	28.86	95.29
EKI-BART (Fan et al., 2020)	-	-	-	35.945	-	<u>16.999</u>	29.583	-
KG-BART (Liu et al., 2021)	-	-	-	33.867	-	16.927	29.634	-
RE-T5 (Wang et al., 2021)	-	-	-	**40.863**	-	**17.663**	**31.079**	-
BART-base-P2T	20.83	42.91	40.74	29.918	30.61	14.670	26.960	92.84
T5-base-P2T	22.38	44.59	44.97	33.577	31.95	16.152	29.104	95.55
BART-large-KW	22.25	43.38	43.87	32.956	32.26	16.065	28.335	96.16
BART-large-Att	22.22	43.80	44.61	33.405	32.03	16.036	28.452	96.43
BART-large-P2T	22.65	43.84	44.78	33.961	32.18	16.174	28.462	96.20
T5-large-KW	23.79	45.73	48.06	37.023	32.85	16.987	29.659	95.32
T5-large-Att	23.94	45.87	47.99	36.947	32.79	16.943	29.607	95.43
T5-large-P2T	23.89	45.77	48.08	<u>37.119</u>	32.94	16.901	<u>29.751</u>	94.82

Table 12: Automatic evaluation results of select SAPPHIRE models on test$_O$ (evaluated by the CommonGen authors). For BART-base and T5-base, we report the best SAPPHIRE model on test$_O$ (P2T), and all three models for BART-large and T5-large. We compare to Lin et al. (2020)'s reported baseline numbers, noting that they did not report BART-base, and published models on their leaderboard[5] that outperform the baselines at the time of writing. Bold corresponds to best performance (for BLEU-4, CIDEr, and SPICE, since their leaderboard only reports these three), and underline corresponds to second best performance.

<u>Model</u>	<u>Method</u>	**Fluency**	**Commonsense**
BART-large	Baseline	3.92	4.06
	Kw-aug	4.13	3.92
	Att-aug	4.10	4.06
	P2T	**4.17**	**4.13**
T5-base	Baseline	4.02	3.83
	Kw-aug	4.04	4.04
	Att-aug	**4.13**	3.98
	P2T	4.02	**4.08**
Human		4.14	4.32

Table 13: Avg. human eval results on test$_{CG}$, rated on 1-5 scales. Bold corresponds to best performance for that model.

8 Conclusion and Future Work

In conclusion, we motivated and proposed several improvements for concept-to-text generation which we call SAPPHIRE: Set Augmentation and Post-hoc PHrase Infilling and REcombination. We demonstrated their effectiveness on CommonGen through experiments on BART and T5. Extensive evaluation showed that SAPPHIRE improves model performance, addresses many issues of the baselines, and has potential for further exploration.

Potential future work includes improving mask infilling performance, and trying combinations of SAPPHIRE techniques as they could be complementary. Better exploiting regularities of CommonGen-like tasks, e.g. invariance to input order, presents another avenue. SAPPHIRE methods can also be investigated for other data-to-text NLG tasks, e.g. WebNLG, and explored for applications such as improving the commonsense reasoning of personalized dialog agents (Li et al., 2020), data augmentation for NLG (Feng et al., 2020, 2021), and constructing pseudo-references for long-context NLG (Gangal et al., 2021b).

Method	**Text**
Concept Set	{sit, chair, toy, hand} (example 1)
BART-base-BL	hands sitting on a chair
BART-base-KW	A boy sits on a chair with a toy in his hand.
BART-base-Att	A child sits on a chair with a toy in his hand.
BART-base-P2T	hands sitting on a chair with toys
BART-base-MI	Children's hands sit on a chair with a toy.
Human	A baby sits on a chair with a toy in one of its hands.
Concept Set	{camel, desert, ride, walk} (example 2)
BART-base-BL	a camel rides a camel in the desert
BART-base-KW	A camel rides down a walkway in the desert.
BART-base-Att	A man is riding camel as he walks through the desert.
BART-base-P2T	A camel rides down a trail in the desert.
BART-base-MI	In the desert, a man rides a camel for a walk.
Human	A loud group of people walk around the desert and ride camels.
Concept Set	{jacket, wear, snow, walk} (example 3)
BART-large-BL	walking in the snow wearing a furry jacket
BART-large-KW	A man wearing a jacket is walking in the snow.
BART-large-Att	A man in a blue jacket is walking in the snow.
BART-large-P2T	A man is wearing a furry jacket as he walks in the snow.
BART-large-MI	A walk in the snow wearing a furry jacket
Human	A man wears a jacket and walks in the snow.
Concept Set	{bench, bus, wait, sit} (example 4)
BART-large-BL	A bus sits on the tracks with people waiting on benches.
BART-large-KW	A bus sits next to a bench waiting for passengers.
BART-large-Att	A woman sits on a bench waiting for a bus.
BART-large-P2T	A bus sits at a stop waiting for passengers to get off the bench.
BART-large-MI	There are people waiting on benches outside bus stops to sit down. pic.twitter.
Human	The man sat on the bench waiting for the bus.
Concept Set	{sunglass, wear, lady, sit} (example 5)
T5-base-BL	A lady sits in a sunglass.
T5-base-KW	A lady sits next to a man wearing sunglasses.
T5-base-Att	A lady sits wearing sunglasses.
T5-base-P2T	A lady sits next to a man wearing sunglasses.
BART-base-MI	A young lady sits in a sunglass to wear.
Human	The lady wants to wear sunglasses, sit, relax, and enjoy her afternoon.
Concept Set	{hold, hand, stand, front} (example 6)
T5-large-BL	Someone stands in front of someone holding a hand.
T5-large-KW	Two men stand in front of each other holding hands.
T5-large-Att	A man stands in front of a woman holding a hand.
T5-large-P2T	A man standing in front of a man holding a hand.
BART-large-MI	Mr. Trump holding a hand to stand in front of
Human	A man stands and holds his hands out in front of him.

Table 14: Qualitative examples for test$_{CG}$. Color coded: baseline, Kw-aug, Att-aug, P2T, MI, and human reference.

Acknowledgments

We thank our anonymous reviewers, Graham Neubig, Ritam Dutt, Divyansh Kaushik, and Zhengbao Jiang for their comments and suggestions.

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.

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

Ricardo Campos, Vítor Mangaravite, Arian Pasquali, A. Jorge, C. Nunes, and A. Jatowt. 2018. Yake! collection-independent automatic keyword extractor. In *ECIR*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Wenchao Du and Alan W Black. 2019. Boosting dialog response generation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 38–43, Florence, Italy. Association for Computational Linguistics.

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

Zhihao Fan, Yeyun Gong, Zhongyu Wei, Siyuan Wang, Yameng Huang, Jian Jiao, Xuanjing Huang, Nan Duan, and Ruofei Zhang. 2020. An enhanced knowledge injection model for commonsense generation. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 2014–2025, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Steven Y. Feng, Varun Gangal, Dongyeop Kang, Teruko Mitamura, and Eduard Hovy. 2020. GenAug: Data augmentation for finetuning text generators. In *Proceedings of Deep Learning Inside Out (DeeLIO): The First Workshop on Knowledge Extraction and Integration for Deep Learning Architectures*, pages 29–42, Online. Association for Computational Linguistics.

Steven Y. Feng, Varun Gangal, Jason Wei, Sarath Chandar, Soroush Vosoughi, Teruko Mitamura, and Eduard Hovy. 2021. A survey of data augmentation approaches for NLP. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 968–988, Online. Association for Computational Linguistics.

Steven Y. Feng, Aaron W. Li, and Jesse Hoey. 2019. Keep calm and switch on! preserving sentiment and fluency in semantic text exchange. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 2701–2711.

Varun Gangal, Steven Y. Feng, Eduard Hovy, and Teruko Mitamura. 2021a. NAREOR: The narrative reordering problem. *arXiv preprint arXiv:2104.06669*.

Varun Gangal, Harsh Jhamtani, Eduard Hovy, and Taylor Berg-Kirkpatrick. 2021b. Improving automated evaluation of open domain dialog via diverse reference augmentation. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 4079–4090, Online. Association for Computational Linguistics.

Claire Gardent, Anastasia Shimorina, Shashi Narayan, and Laura Perez-Beltrachini. 2017. The webnlg challenge: Generating text from rdf data. In *Proceedings of the 10th International Conference on Natural Language Generation*, pages 124–133.

Deepanway Ghosal, Navonil Majumder, Alexander Gelbukh, Rada Mihalcea, and Soujanya Poria. 2020. COSMIC: COmmonSense knowledge for eMotion identification in conversations. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 2470–2481, Online. Association for Computational Linguistics.

Maarten Grootendorst. 2020. Keybert: Minimal keyword extraction with bert.

Harsh Jhamtani, Varun Gangal, Eduard Hovy, Graham Neubig, and Taylor Berg-Kirkpatrick. 2018. Learning to generate move-by-move commentary for chess games from large-scale social forum data. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1661–1671.

Chloé Kiddon, Luke Zettlemoyer, and Yejin Choi. 2016. Globally coherent text generation with neural checklist models. In *Proceedings of the 2016 conference on empirical methods in natural language processing*, pages 329–339.

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

Aaron W. Li, Veronica Jiang, Steven Y. Feng, Julia Sprague, Wei Zhou, and Jesse Hoey. 2020. ALOHA: Artificial learning of human attributes for dialogue agents. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(05):8155–8163.

Jiwei Li, Michel Galley, Chris Brockett, Jianfeng Gao, and Bill Dolan. 2016. A diversity-promoting objective function for neural conversation models. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 110–119, San Diego, California. Association for Computational Linguistics.

Percy Liang, Michael I 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*, pages 91–99.

Bill Yuchen Lin, Wangchunshu Zhou, Ming Shen, Pei Zhou, Chandra Bhagavatula, Yejin Choi, and Xiang Ren. 2020. CommonGen: A constrained text generation challenge for generative commonsense reasoning. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 1823–1840, Online. Association for Computational Linguistics.

Ye Liu, Yao Wan, Lifang He, Hao Peng, and Philip S. Yu. 2021. KG-BART: Knowledge graph-augmented bart for generative commonsense reasoning. *Proceedings of the AAAI Conference on Artificial Intelligence*, 35(7):6418–6425.

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.

Sebastien Montella, Betty Fabre, Tanguy Urvoy, Johannes Heinecke, and Lina Rojas-Barahona. 2020. Denoising pre-training and data augmentation strategies for enhanced RDF verbalization with transformers. In *Proceedings of the 3rd International Workshop on Natural Language Generation from the Semantic Web (WebNLG+)*, pages 89–99, Dublin, Ireland (Virtual). Association for Computational Linguistics.

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

Edwin JG Pitman. 1937. Significance tests which may be applied to samples from any populations. *Supplement to the Journal of the Royal Statistical Society*, 4(1):119–130.

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.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140):1–67.

Alon Talmor, Yanai Elazar, Yoav Goldberg, and Jonathan Berant. 2020. oLMpics-on what language model pre-training captures. *Transactions of the Association for Computational Linguistics*, 8:743–758.

Shubhangi Tandon, TS Sharath, Shereen Oraby, Lena Reed, Stephanie Lukin, and Marilyn Walker. 2018. TNT-NLG, System 2: Data repetition and meaning representation manipulation to improve neural generation. *E2E NLG Challenge System Descriptions*.

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.

Han Wang, Yang Liu, Chenguang Zhu, Linjun Shou, Ming Gong, Yichong Xu, and Michael Zeng. 2021. Retrieval enhanced model for commonsense generation. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 3056–3062, Online. Association for Computational Linguistics.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q Weinberger, and Yoav Artzi. 2020. BERTScore: Evaluating text generation with BERT. In *International Conference on Learning Representations 2020*.

Appendices

A Model Training and Generation Details

T5-large consists of 770M params, T5-base 220M params, BART-large 406M params, and BART-base 139M params. We train two seeded versions of each baseline model and SAPPHIRE model. For all models, we use beam search with a beam size of 5, decoder early stopping, a decoder length penalty of 0.6, encoder and decoder maximum lengths of 32, and a decoder minimum length of 1. For model training, we use a batch size of 128 for T5-base and BART-base, 32 for BART-large, and 16 for T5-large. For T5-base, T5-large, and BART-base, we use 400 warmup steps, and 500 for BART-large. We train all models up to a reasonable number of epochs (e.g. 10 or 20) and perform early stopping using our best judgment (e.g. if metrics have continually decreased for multiple epochs). The learning rates for SAPPHIRE models were determined by trying a range of values (e.g. from 1e-6 to 1e-4), and finding ones which led to good convergence behavior (e.g. validation metrics increase at a decently steady rate and reach max. after a reasonable number of epochs). For the best-performing models, learning rates are as follows (each set consists of {baseline,Kw-aug,Att-aug,P2T}): BART-base = {3e-05,2e-05,3e-05,1e-05}, BART-large = {3e-05,2e-05,2e-05,5e-06}, T5-base = {5e-05,5e-05,5e-05,1e-05}, T5-large = {2e-05,2e-05,2e-05,5e-06}.

Training was done using single RTX 2080 Ti and Titan Xp GPUs, and Google Colab instances which alternately used a single V100, P100, or Tesla T4 GPU. The vast majority of the training was done on a single V100 per model. T5-base models trained in approx. 1 hour, BART-base models in approx. 45 minutes, T5-large models in approx. 4 hours, and BART-large models in approx. 1.5-2 hours.

B Full Re-implementation versus Reported Model Numbers

See Table 16 for full comparison of our re-implemented CommonGen models compared to the original reported numbers in Lin et al. (2020).

C Human Evaluation Details

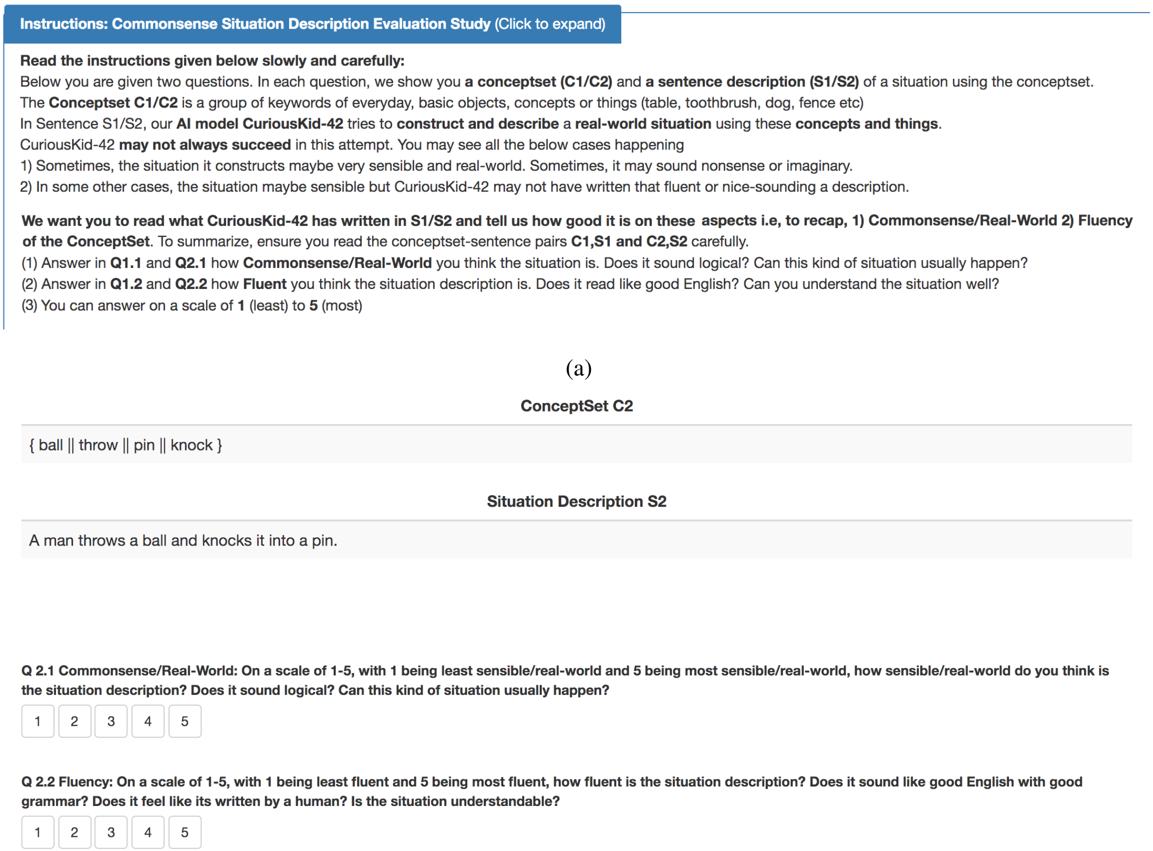
Human evaluation was done via paid crowdworkers on AMT, who were from Anglophone countries with lifetime approval rates > 97% . Each example was evaluated by 2 annotators. The time given for each AMT task instance or HIT was 8 minutes. Sufficient time to read instructions, as calibrated by authors, was also considered in the maximum time limit for performing each HIT. Annotators were paid 98 cents per HIT. This rate (7.35$/hr) exceeds the minimum wage for the USA (7.2$/hr) and constitutes fair pay. We neither solicit, record, request, or predict any personal information pertaining to the AMT crowdworkers. Specific instructions and a question snippet can be seen in Figure 1.

Method	**Text**
Concept Set	{food, eat, hand, bird}
BART-base-BL	hands of a bird eating food
BART-base-KW	a bird eats food from a hand
BART-base-Att	hand of a bird eating food
BART-base-P2T	A bird is eating food with its hand.
BART-base-MI	The food is in the hands of a bird eating it.
Human	A small bird eats food from someone's hand.
Concept Set	{front, dance, routine, perform}
BART-base-BL	A woman performs a routine in front of a dance.
BART-base-KW	A man performs a routine in front of a group of people.
BART-base-Att	A man is performing a routine in front of a group of people.
BART-base-P2T	A woman performs a routine in front of a group of people.
BART-base-MI	In this dance, a man performs a routine in front of a mirror.
Human	The girl performed her dance routine in front of the audience.
Concept Set	{chase, ball, owner, dog, throw}
BART-base-BL	A dog is throwing a ball into a chase.
BART-base-KW	A dog is about to throw a ball to its owner.
BART-base-Att	A dog is trying to throw a ball at its owner.
BART-base-P2T	A dog is chasing the owner of a ball.
BART-base-MI	The dog was trained to throw balls and the dog would chase after the owner.
Human	The owner threw the ball for the dog to chase after.
Concept Set	{music, dance, room, listen}
BART-large-BL	A listening music and dancing in a dark room
BART-large-KW	A group of people dance and listen to music in a room.
BART-large-Att	A group of people are dancing and listening to music in a room.
BART-large-P2T	Two people are dancing and listening to music in a dark room.
BART-large-MI	Music and dancing in the dance floor.
Human	A boy danced around the room while listening to music.
Concept Set	{cheer, team, crowd, goal}
T5-base-BL	the crowd cheered after the goal.
T5-base-KW	the crowd cheered after the goal by football team
T5-base-Att	the crowd cheered after the goal by the team.
T5-base-P2T	the crowd cheered as the team scored their first goal.
BART-base-MI	The team and the crowd cheered after the goal.
Human	The crowd cheered when their team scored a goal.
Concept Set	{bag, put, apple, tree, pick}
T5-base-BL	A man puts a bag of apples on a tree.
T5-base-KW	A man puts a bag under a tree and picks an apple.
T5-base-Att	A man puts a bag under a tree and picks an apple.
T5-base-P2T	A man puts a bag of apples on a tree and picks them.
BART-base-MI	A man puts a bag of apple juice on a tree to pick it up
Human	I picked an apple from the tree and put it in my bag.
Concept Set	{circle, ball, throw, turn, hold}
T5-large-BL	Someone turns and throws a ball in a circle.
T5-large-KW	A man holds a ball and turns to throw it into a circle.
T5-large-Att	A man holds a ball in a circle and throws it.
T5-large-P2T	A man holds a ball, turns and throws it into a circle.
BART-large-MI	He turns and throws a ball into the circle to hold it.
Human	A girl holds the ball tightly, then turns to the left and throws the ball into the net which is in the shape of a circle.
Concept Set	{hair, sink, lay, wash}
T5-large-BL	A woman is washing her hair in a sink.
T5-large-KW	A woman lays down to wash her hair in a sink.
T5-large-Att	A man lays down to wash his hair in a sink.
T5-large-P2T	A woman is washing her hair in a sink.
BART-large-MI	A woman is washing her hair in the sink. She lay the sink down
Human	The woman laid back in the salon chair, letting the hairdresser wash her hair in the sink.
Concept Set	{wash, dry, towel, face}
T5-large-BL	A man is washing his face with a towel.
T5-large-KW	A man washes his face with a towel and then dries it.
T5-large-Att	A man is washing his face with a towel and drying it.
T5-large-P2T	A man is washing his face with a towel and drying it off.
BART-large-MI	A man is washing his face with a towel to dry it.
Human	The woman will wash the baby's face and dry it with a towel.

Table 15: Qualitative examples for test$_{CG}$. Color coded: baseline, Kw-aug, Att-aug, P2T, MI, and human reference.

Model\Metrics	ROUGE-2/L		BLEU-3/4		METEOR	CIDEr	SPICE	BERTScore	Cov
Reported BART-large	22.13	43.02	37.00	27.50	31.00	14.12	30.00	-	97.56
Reported T5-base	15.33	36.20	28.10	18.00	24.60	9.73	23.40	-	83.77
Reported T5-Large	21.98	44.41	40.80	30.60	31.00	15.84	31.80	-	97.04
Our BART-base	15.91	36.15	38.30	28.30	30.20	15.07	30.35	58.26	93.44
Our BART-large	17.27	37.32	**39.95**	**30.20**	**31.15**	**15.72**	**31.20**	58.58	95.03
Our T5-base	**17.27**	**37.69**	**41.15**	**31.00**	**31.10**	**16.37**	**32.05**	60.32	**94.44**
Our T5-large	17.90	38.31	**43.80**	**33.60**	**32.70**	**17.02**	**33.45**	61.39	96.26

Table 16: Performance of our re-implemented CommonGen models on dev$_O$ compared to the original numbers reported in Lin et al. (2020). Note that for our models, results are averaged over two seeds, and that the original authors did not experiment with BART-base. Bold indicates where we match or exceed the reported metric.



Instructions: Commonsense Situation Description Evaluation Study (Click to expand)

Read the instructions given below slowly and carefully:

Below you are given two questions. In each question, we show you **a conceptset (C1/C2)** and **a sentence description (S1/S2)** of a situation using the conceptset.

The **Conceptset C1/C2** is a group of keywords of everyday, basic objects, concepts or things (table, toothbrush, dog, fence etc)

In Sentence S1/S2, our **AI model CuriousKid-42** tries to **construct and describe** a **real-world situation** using these **concepts and things**.

CuriousKid-42 **may not always succeed** in this attempt. You may see all the below cases happening

1) Sometimes, the situation it constructs maybe very sensible and real-world. Sometimes, it may sound nonsense or imaginary.

2) In some other cases, the situation maybe sensible but CuriousKid-42 may not have written that fluent or nice-sounding a description.

We want you to read what CuriousKid-42 has written in S1/S2 and tell us how good it is on these aspects i.e, to recap, 1) Commonsense/Real-World 2) Fluency of the ConceptSet. To summarize, ensure you read the conceptset-sentence pairs **C1,S1 and C2,S2** carefully.

(1) Answer in **Q1.1** and **Q2.1** how **Commonsense/Real-World** you think the situation is. Does it sound logical? Can this kind of situation usually happen?

(2) Answer in **Q1.2** and **Q2.2** how **Fluent** you think the situation description is. Does it read like good English? Can you understand the situation well?

(3) You can answer on a scale of **1** (least) to **5** (most)

(a)

ConceptSet C2

{ ball || throw || pin || knock }

Situation Description S2

A man throws a ball and knocks it into a pin.

Q 2.1 Commonsense/Real-World: On a scale of 1-5, with 1 being least sensible/real-world and 5 being most sensible/real-world, how sensible/real-world do you think is the situation description? Does it sound logical? Can this kind of situation usually happen?

□ 1 □ 2 □ 3 □ 4 □ 5

Q 2.2 Fluency: On a scale of 1-5, with 1 being least fluent and 5 being most fluent, how fluent is the situation description? Does it sound like good English with good grammar? Does it feel like its written by a human? Is the situation understandable?

□ 1 □ 2 □ 3 □ 4 □ 5

(b)

Figure 1: Snapshots of human evaluation: a) instructions seen by annotator and b) an example with questions.

D Graphs Displaying Other Hyperparameter Results

Figures 2, 3, 4, and 5 contain graphs displaying other hyperparameter results for Kw-aug, Att-aug, P2T, and Mask Infilling (MI), respectively.

E Further Qualitative Examples

See Table 15 for further qualitative examples.

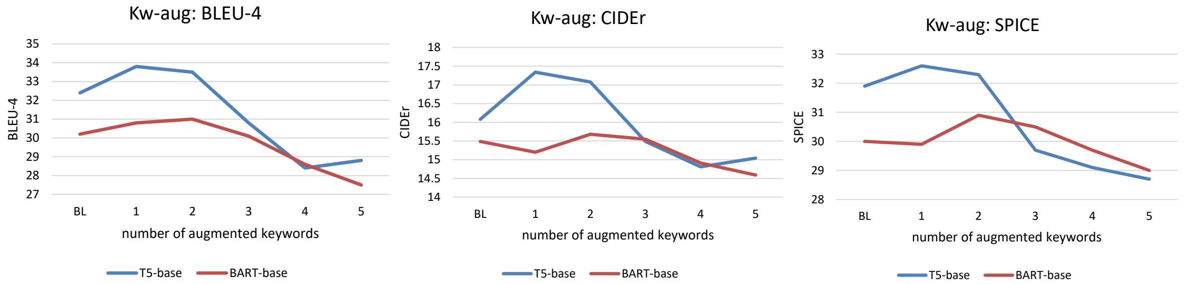


Figure 2: Kw-aug: graphs of BLEU-4, CIDEr, and SPICE results on test$_{CG}$ over different numbers of augmented keywords for BART-base and T5-base. These are only first seed results, and we only went above three augmented keywords for the base size models. BL refers to the baseline results with no augmented keywords.

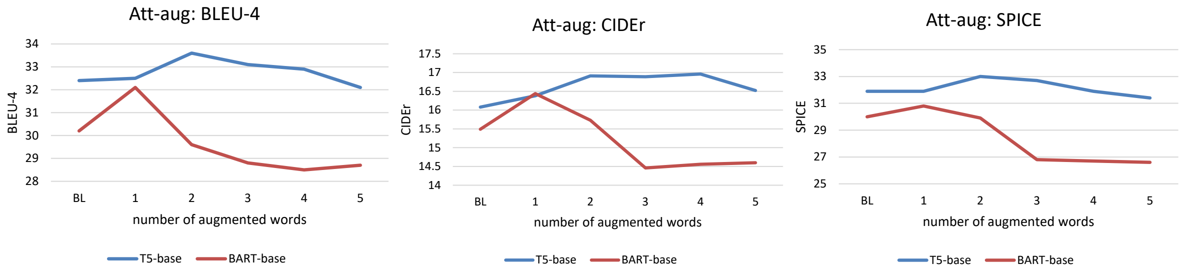


Figure 3: Att-aug: graphs of BLEU-4, CIDEr, and SPICE results on test$_{CG}$ over different numbers of augmented words for BART-base and T5-base. These are only first seed results, and we only went above three augmented words for the base size models. BL refers to the baseline results with no augmented words.

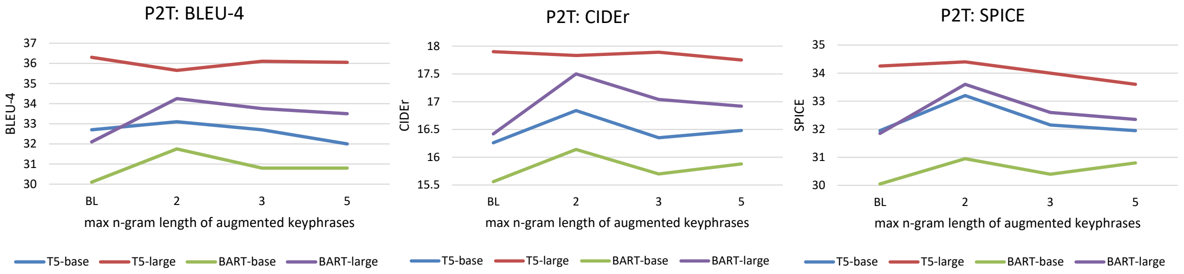


Figure 4: P2T: graphs of BLEU-4, CIDEr, and SPICE results on test$_{CG}$ over different max n-gram lengths of augmented keyphrases. These are results averaged over two seeds. BL refers to the baseline results with no augmented keyphrases.

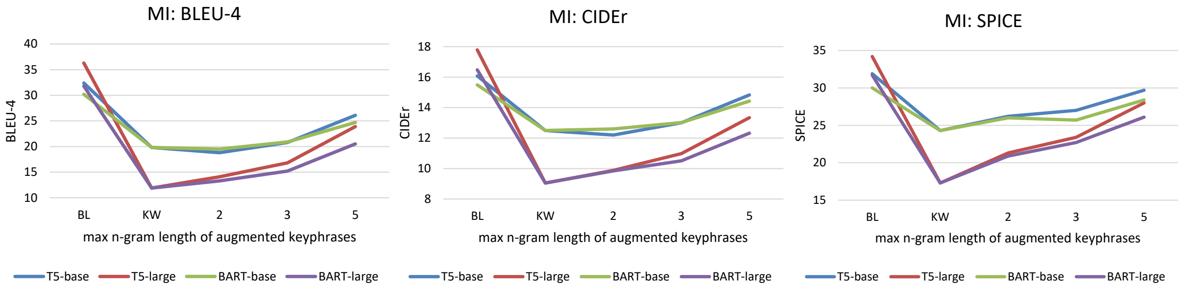


Figure 5: Mask Infilling (MI): graphs of BLEU-4, CIDEr, and SPICE results on test$_{CG}$ over different max n-gram lengths of augmented keyphrases. These are first seed results only. BL refers to the baseline results, and KW refers to mask infilling on the original keywords only (with no augmented keyphrases).