The Case for a Single Model that can Both Generate Continuations and Fill in the Blank

Daphne Ippolito[1,2] **Liam Dugan**[1] **Emily Reif**[2]
Ann Yuan[2] **Andy Coenen**[2] **Chris Callison-Burch**[1]
[1]University of Pennsylvania [2]Google Research
{daphnei, ldugan, ccb}@seas.upenn.edu
{ereif, annyuan, andycoenen}@google.com

Abstract

The task of inserting text into a specified position in a passage, known as fill in the blank (FITB), is useful for a variety of applications where writers interact with a natural language generation (NLG) system to craft text. While previous work has tackled this problem with models trained specifically to do the fill-in-the-blank task, a more useful model is one that can effectively perform *both* FITB and continuation. In this work, we evaluate the feasibility of using a single model to do both tasks. We show that models pre-trained with a FITB-style objective are capable of both tasks, while models pre-trained for continuation are not. Finally, we show how FITB models can be easily finetuned to allow for fine-grained control over the length and word choice of the generation.

1 Introduction

Natural language generation systems are increasingly being incorporated into applications where a human writer and an AI jointly collaborate to construct text. These range from creative domains such as collaborative story writing (Coenen et al., 2021; Akoury et al., 2020) to more practical ones such as email composition and code synthesis (Buschek et al., 2021; Wu, 2018; Austin et al., 2021). These applications are often limited to generating text at the end of what has been written so far. This is because language models (LMs) are typically designed to produce text by repeatedly predicting the next word in a sequence given the previous words. However, there is a need for more powerful interactive tools which enable writers to solicit insertions at any chosen position within the existing text, a task referred to as fill in the blank (FITB) or infilling. For example, a creative writer might want a tool which can insert a description of a place or character, and a programmer might want a system that can fill in the contents of a function located in the middle of their code.

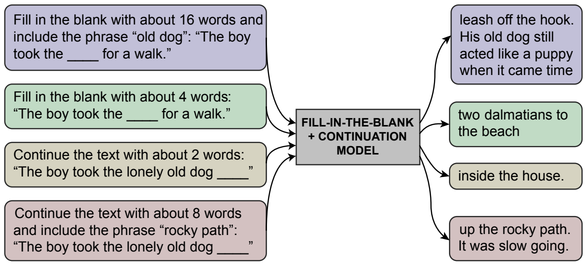

Figure 1: A single model that can handle a variety of related writing tasks is more efficient than separate models per task.

Most prior work tackling FITB consider it a separate task from continuation, one to be specifically optimized for, for example training a model from scratch (Ippolito et al., 2019; Zhu et al., 2019) or finetuning a model trained originally for continuation (Donahue et al., 2020). However, having separate trained models for FITB and for continuation is inefficient for downstream applications where maintaining multiple neural networks can be prohibitive.

Any model that can do FITB can be made to do continuation simply by placing the blank at the end of the input. Thus, in this work we describe how models trained on FITB can be employed effectively for both infilling and continuation operations. We show how T5 (Raffel et al., 2020), one of the most popular pre-trained models, can reasonably handle both tasks, as it was pre-trained with a FITB-like objective. Finetuning T5 further improves its ability and also allows for the incorporation of controllability of generation length and word choice.

2 Supporting FITB and Continuation

Definitions. We define filling in the blank as the task of predicting text to replace a single missing span, usually demarcated with a special token, in an input text passage. (Some prior work considers inputs with multiple blanks, but inserting text at one position at a time better matches the kinds of edits humans do.) We define continuation in the

Findings of the Association for Computational Linguistics: NAACL 2022, pages 2421 - 2432
July 10-15, 2022 ©2022 Association for Computational Linguistics

Example Type	Input	Target
C4FILLBLANK no goal	fill: I love avocados. I ate a sandwich covered in them. **_8_** I talked to my doctor about it later. It turned out I was allergic to avocados.	After I ate it, my mouth was itchy and tingly.
C4FILLBLANK with goal	fill: I love avocados. I ate a sandwich covered in them. **_8_** I talked to my doctor about it later. It turned out I was allergic to avocados. **Goal: mouth was itchy**	After I ate it, my mouth was itchy and tingly.
C4FILLBLANK no goal	fill: I love avocados. I ate a sandwich covered in them. After I ate it, my mouth was itchy and tingly. I talked to my doctor about it later. **_8_**	It turned out I was allergic to avocados.
C4FILLEND with goal	fill: I love avocados. I ate a sandwich covered in them. After I ate it, my mouth was itchy and tingly. I talked to my doctor about it later. **_8_ Goal: allergic to**	It turned out I was allergic to avocados.

Table 1: Examples of the finetuning objectives. "8" is the approximate length in words of the target sequence. During finetuning, about 25% of training examples took each of these formats.

traditional language modeling sense–predicting the next token in a sequence given only the previous tokens. Donahue et al. (2020) discuss how language modeling is a special case of infilling, and they use this as justification to finetune a continuation-based language model to do infilling. However, we argue that if continuation is a subtask of infilling, it makes more sense to go in the opposite direction: prioritize a model which can do infilling and check that it achieves satisfactory performance at continuation.

Using a model pre-trained for FITB. T5 is a model pre-trained with a "span corruption" objective very similar to FITB; the model is asked to reconstruct the missing text after random subsequences of the input are replaced with special identifiers. Thus, a pre-trained T5 model can be used without any further training to do both continuation and infilling by appropriately choosing text to mask out. The encoder-decoder architecture of T5 is also more conducive to FITB than the decoder-only architectures that are typically used for continuation-based generation, such as GPT-2 (Radford et al., 2019). This is because the attention mechanism in encoder-decoder architectures allows the context on the left side of the blank to attend to the context on the right, while decoder-only architectures only support masked attention (each token can only attend to the positions to its left).

Even though T5's pre-training objective was a form of FITB, finetuning is still advantageous. For one, our definition of FITB involves only a single masked out substring, not multiple, so finetuning improves alignment with the goal task. Finetuning also allows us to incorporate additional conditioning signals not supported by the pre-trained T5, such as being able to specify the desired length of the generated text or specify words that ought to be included in the blank, a task we refer to as "goal conditioning." Length control, which comes by default in a traditional language model by simply sampling more or fewer tokens, is particularly necessary for FITB, where the end of the generation must fit seamlessly with the text to its right.

Using a model pre-trained for continuation. The biggest language models available today were largely trained in the continuation rather than the FITB paradigm (Brown et al., 2020b; Black et al., 2021). Since our primary goal is to have a single model for both tasks, we also address the question: if a continuation-trained model is big enough, can it handle FITB without the need for finetuning? Few-shot learning with large language models, as popularized by Brown et al. (2020b), has had success on many tasks in NLP. We try out this approach for FITB by designing a few-shot prompt containing several demonstrations of the FITB task, formulated in a similar "infilling by language modelling" template to that proposed by Donahue et al. (2020). Further details on our approach to selecting a few-shot prompt are in Appendix A.1.

3 Experiments

Model. For all experiments with T5, we use the 800M parameter v1.1 'large' model (Appendix A.4 gives additional results from the 3B parameter 'XL' model). To finetune T5 for FITB, we construct training examples from documents by first partitioning the document text into a left context, gap, and right context. The input to the model is then the left and right contexts concatenated with textual representations of the additional conditioning signals. The target sequence is the true text for the blank. This formulation easily supports continuation, as the blank can be deliberately placed at the end (i.e., providing no right context). Finetuning examples are drawn from C4, the same dataset T5 was pre-trained on. Documents are split into word sequences, and these are then randomly truncated to be between 256 and 512 words long. A substring of between 1 and 64 words is selected to be blanked out. For half of the finetuning examples, the location of the blank is randomly selected, and for the other half, it is always placed at the end. To support length conditioning, we follow Roberts and Raffel (2020) and include a bucketed version of the target length as part of the blank. To support goal conditioning, for half the examples, a random substring of up to half the words of the target is appended to the end of the input. Examples are shown in Table 1.

Baselines We compare T5 against Thoppilan et al. (2022)'s 137B parameter decoder-only language model (referred to in this paper as LLM). Trained explicitly for continuation, this model has been used successfully for few-shot learning in other domains (Austin et al., 2021; Reif et al., 2021). We use the LLM in two ways: (1) as a standard continuation model, prompting with only the left context of an example; and (2) in a few-shot learning paradigm.

Evaluation Datasets We evaluate continuation and FITB on C4 as well as two story writing datasets. We chose this domain because creative writing assistant applications are one of the key areas we expect to benefit from multi-task models (Coenen et al., 2021). Reddit Writing Prompts (RWP) is a corpus of stories from the 'r/WritingPrompts' sub-Reddit (Fan et al., 2018), and we construct validation sets RWPFILLBLANK and RWPFILLEND using the same method described in the previous section. We cap the C4 and RWP validation sets to 5,000 examples each. ROC Stories (ROC) is a crowd-sourced dataset of five-sentence commonsense stories (Mostafazadeh et al., 2016). For ROC Stories, the 2018 validation set is used to construct ROCFILLMIDDLE, where the middle sentence of each story is blanked out, and ROCFILLEND, where the last sentence is blanked out. Unless otherwise noted, all evaluation is done without goal conditioning and uses random sampling with top-k=50 as the decoding strategy. Example generations for all evaluation sets can be found at `https://bit.ly/2U0Ixxa`.

4 Findings

Automatic Evaluation We measure the fluency of proposed generations by evaluating the perplexity of each dataset's examples when the predicted text is placed in the blank (Donahue et al., 2020).

	C4FILL BLANK	RWPFILL MIDDLE	ROCFILL BLANK
Few-shot LLM	14.14	19.48	18.21
Pre-trained T5	10.38	14.08	22.62
Finetuned T5	10.33	14.08	20.47
Donahue et al. (2020)	N/A	N/A	23.28
Groundtruth	9.41	12.99	16.90

Table 2: Perplexity of evaluation sets according to LLM when the blank has been filled with approaches involving no fine-tuning (top), finetuned approaches (middle), and the groundtruth (bottom). Lower values indicate that the text was considered more fluent by the LLM.

	C4FILL END	RWPFILL END	ROCFILL END
LLM	9.34	12.82	15.55
Pre-trained T5	10.09	13.51	21.71
T5 FILLBLANKCONT	10.04	13.74	19.60
T5 LM-ADAPTION	10.06	13.71	19.68
Groundtruth	9.41	12.99	16.90

Table 3: Perplexity of continuation-based evaluation sets when a continuation has been generated using approaches with no finetuning (top) and two settings of finetuning T5 (middle).

We use the LLM to measure perplexity[1]. The results are shown in Table 2. We see that the LLM struggles to generate fluent infills, even when used in a few-shot setting. The only exception to this is ROC Stories, a dataset with fairly simplistic, predictable language. Finetuning T5 does not result in significantly improved fluency over the pre-trained model except on ROC Stories. Lastly, for ROC Stories, we compare against Donahue et al. (2020)'s finetuned GPT-2 small, which yielded less fluent predictions.

Table 3 shows a similar analysis on our continuation-style datasets. We see that the pre-trained T5 generates about as fluent continuations as T5 finetuned in the manner described in Section 3 (T5 FILLBLANKCONT), as well as T5 finetuned for the same number of steps, but only on the continuation task (T5 LM-ADAPTION). The first row of Table 3 shows how fluent the LLM scores its own generated continuations.

Human Evaluation Human evaluation was conducted on 70 examples, 35 from RWPFILLBLANK and 35 from RWPFILLEND, with examples about evenly distributed across length buckets. For RWPFILLBLANK evaluation tasks, the rater was presented an input context and several possible sequences that could go in the blank. They were

[1]Note, since this is the same model being used for generation for our continuation baseline, this metric may be biased.

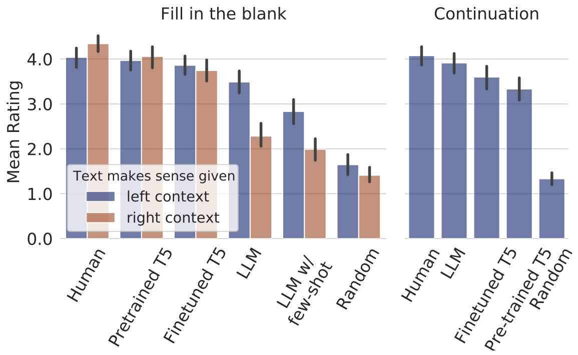

Figure 2: Human ratings of FITB generations (left) and continuation generations (right). Error bars are 95% confidence intervals.

Finetuned T5	Context	Length
C4FILLBLANK	0.860	0.877
RWPFILLBLANK	0.797	0.881
C4FILLEND	0.858	0.775
RWPFILLEND	0.791	0.746

Table 4: Accuracy of models finetuned on FILL-BLANKCONT at correctly using provided length and goal conditioning signals.

asked to rate each sequence first, on how well it fit the text before it, and second, on how well it fit with the text following it, according to a 5-point slider. For RWPFILLEND, the task was almost the same, except that the rater was presented only a left context and asked to rate how well it continued the prompt. More details are in Appendix A.3. Figure 2 shows the results.

On the FITB task, the pre-trained and finetuned T5 models were indistinguishable in terms of quality. The LLM that formed continuations prompted with only the left context did somewhat better than the few-shot LLM, indicating that few-shot learning is not yet a feasible alternative to finetuning. On the continuation task, the LLM has the highest rating, which is unsurprising since it is a much larger model than T5. However, the finetuned T5 is rated almost as highly. Overall, these results suggest that T5, unlike the LLM, can be used effectively for continuation as well as FITB. Furthermore, if one doesn't care about controllability, pre-trained T5 can be used effectively for both tasks without any further finetuning.

Benefits of Controllability Despite finetuning not resulting in significantly more fluent outputs, there are still good reasons to finetune; namely, finetuning allows for increased controllability. For example, length conditioning is extremely important for FITBmodels, since it is not possible to control the generation length by simply sampling more or fewer tokens. Pre-trained T5 tends to produce infill proposals which are shorter than the groundtruth (Figure 3), and there is no way to ask the model to produce longer generations. In contrast, finetuned T5 was able to produce generations in the target length bucket over 74% of the time (Table 4). Goal conditioning, while not strictly necessary for either task, has been shown to be useful for generative commonsense reasoning (Lin et al., 2020) and may empower users in downstream applications such as AI-assisted creative writing (Roemmele, 2021). Finetuned T5 is able to use all of the specified goal words over 79% of the time.

Domain Transfer Prior work on FITB tends to only evaluate models trained on data from the same domain as the validation set. Our results show that despite training exclusively on C4, T5 models have strong transferability to more targeted domains such as Reddit Writing Prompts. This sort of transferability is extremely important for achieving the goal of having a single model which can handle many tasks and domains.

5 Related Work

FITB is a form of Cloze task (Taylor, 1953). Prior deep-learning approaches to this task include training an encoder-decoder model from scratch with length and goal word conditioning (Ippolito et al., 2019); finetuning GPT-2 (Radford et al., 2019; Donahue et al., 2020); and training a custom self-attention architecture on corrupted text (Zhu et al., 2019). None of these show that their fill-in-the-blank models remain effective at continuation or perform well on text domains that differ from the training data. Related to FITB, Mori et al. (2020) investigate a setting where a sentence is randomly deleted from the input, and the model must both predict the location of the deletion as well as its contents. Huang et al. (2020) tackle the sentence infilling task using a mixture of BERT and GPT-2. Lastly, many LM pre-training objectives involve masking out parts of the input then predicting the masked values, which is similar to FITB (Devlin et al., 2019; Raffel et al., 2020; Joshi et al., 2020).

6 Conclusion

In this work, we make the case for starting with a model capable of filling in the blank when attempting to build a system that can perform both FITB

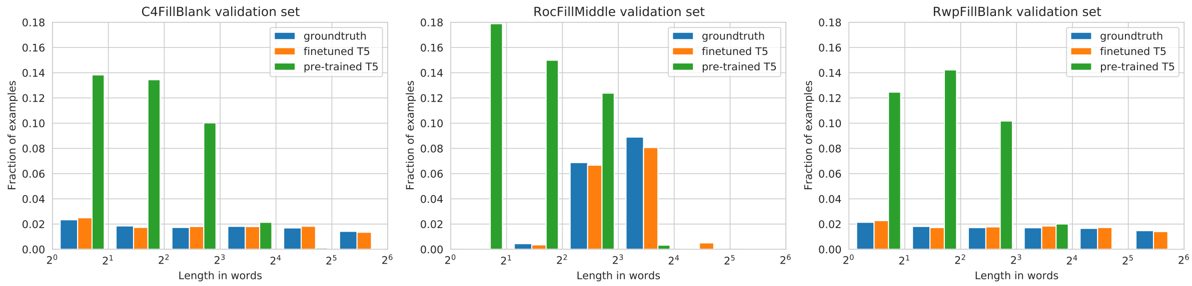

Figure 3: For each of the FITB validation sets, a histogram of the distribution of sequence lengths (measured in words) of the ground-truth blanked out text and the proposed infills from T5 (after and before finetuning). We see that pre-trained T5 tends to produce text that is shorter than the groundtruth.

and continuation. As LMs become bigger, it will be unsustainable to have separately trained models for each generation task. Multi-task, domain-transferable models, such as the ones we propose, require less total training and are more efficient to store and use at inference time. While pre-trained T5 by itself is capable of both infilling and continuation, additional conditioning signals such as desired length and goal text can be successfully incorporated into fine-tuning in order to support an even greater diversity of model interactions. We focused our experiments on the T5 model; however, we expect that other model families and architectures can be trained similarly to support a variety of generation tasks. For example, GPT-3 (Brown et al., 2020a) recently began supporting "insertion" and "edit" interactions in addition to continuation. Finally, we present a negative result that while few-shot learning is a promising method for building multi-task support without any finetuning, it is challenging to make work for the FITB task.

7 Risks and Limitations

All neural language models, including the ones used in this paper, reflect the biases and other issues present in their training data. Weidinger et al. (2021) discuss these risks in detail. The models and datasets considered in this paper are all in the English, and the proposed methods may work differently in other languages. In addition, the paper mostly focuses on showing results pertinent to the story writing domain; in other domains joint models for continuation and fill-in-the-blank might work worse. Finally, the LLMused in this paper is not publicly available, which to some extent limits its reproducibility, though we expect our findings would have been similar had we evaluated with a public model such as GPT-2. We emphasize that the main contribution of this paper is a comparison of different methods, all of which are easily implementable, rather than new model checkpoints.

References

Nader Akoury, Shufan Wang, Josh Whiting, Stephen Hood, Nanyun Peng, and Mohit Iyyer. 2020. Storium: A dataset and evaluation platform for machine-in-the-loop story generation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6470–6484.

Jacob Austin, Augustus Odena, Maxwell Nye, Maarten Bosma, Henryk Michalewski, David Dohan, Ellen Jiang, Carrie Cai, Michael Terry, Quoc Le, et al. 2021. Program synthesis with large language models. *arXiv preprint arXiv:2108.07732*.

Sid Black, Leo Gao, Phil Wang, Connor Leahy, and Stella Biderman. 2021. GPT-Neo: Large Scale Autoregressive Language Modeling with Mesh-Tensorflow.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020a. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020b. Language models are few-shot learners.

Daniel Buschek, Martin Zürn, and Malin Eiband. 2021. The impact of multiple parallel phrase suggestions on email input and composition behaviour of native and non-native english writers. In *Proceedings* *of the 2021 CHI Conference on Human Factors in Computing Systems*, pages 1–13.

Andy Coenen, Luke Davis, Daphne Ippolito, Ann Yuan, and Emily Reif. 2021. Wordcraft: a human-ai collaborative editor for story writing.

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.

Chris Donahue, Mina Lee, and Percy Liang. 2020. Enabling language models to fill in the blanks. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 2492–2501.

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.

Yichen Huang, Yizhe Zhang, Oussama Elachqar, and Yu Cheng. 2020. Inset: Sentence infilling with inter-sentential transformer. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 2502–2515.

Daphne Ippolito, David Grangier, Chris Callison-Burch, and Douglas Eck. 2019. Unsupervised hierarchical story infilling. In *Proceedings of the First Workshop on Narrative Understanding*, pages 37–43.

Mandar Joshi, Danqi Chen, Yinhan Liu, Daniel S Weld, Luke Zettlemoyer, and Omer Levy. 2020. Spanbert: Improving pre-training by representing and predicting spans. *Transactions of the Association for Computational Linguistics*, 8:64–77.

Xiang Lisa Li and Percy Liang. 2021. Prefix-tuning: Optimizing continuous prompts for generation. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 4582–4597.

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 *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: Findings*, pages 1823–1840.

Yusuke Mori, Hiroaki Yamane, Yusuke Mukuta, and Tatsuya Harada. 2020. Finding and generating a missing part for story completion. In *Proceedings of the The 4th Joint SIGHUM Workshop on Computational Linguistics for Cultural Heritage, Social Sciences, Humanities and Literature*, pages 156–166.

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.

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.

Emily Reif, Daphne Ippolito, Ann Yuan, Andy Coenen, Chris Callison-Burch, and Jason Wei. 2021. A recipe for arbitrary text style transfer with large language models. *arXiv preprint arXiv:2109.03910*.

Adam Roberts and Colin Raffel. 2020. Exploring transfer learning with t5: The text-to-text transfer transformer. *Google AI Blog*.

Melissa Roemmele. 2021. Inspiration through observation: Demonstrating the influence of automatically generated text on creative writing. pages 52–461.

Wilson L Taylor. 1953. "cloze procedure": A new tool for measuring readability. *Journalism Bulletin*, 30(4):415–433.

Romal Thoppilan, Daniel De Freitas, Jamie Hall, Noam Shazeer, Apoorv Kulshreshtha, Heng-Tze Cheng, Alicia Jin, Taylor Bos, Leslie Baker, Yu Du, et al. 2022. Lamda: Language models for dialog applications. *arXiv preprint arXiv:2201.08239*.

Laura Weidinger, John Mellor, Maribeth Rauh, Conor Griffin, Jonathan Uesato, Po-Sen Huang, Myra Cheng, Mia Glaese, Borja Balle, Atoosa Kasirzadeh, Zac Kenton, Sasha Brown, Will Hawkins, Tom Stepleton, Courtney Biles, Abeba Birhane, Julia Haas, Laura Rimell, Lisa Anne Hendricks, William Isaac, Sean Legassick, Geoffrey Irving, and Iason Gabriel. 2021. Ethical and social risks of harm from language models. *arXiv preprint arXiv:2112.04359*.

Yonghui Wu. 2018. Smart compose: Using neural networks to help write emails. *Google AI Blog*.

Zihao Zhao, Eric Wallace, Shi Feng, Dan Klein, and Sameer Singh. 2021. Calibrate before use: Improving few-shot performance of language models. In *International Conference on Machine Learning*, pages 12697–12706. PMLR.

Wanrong Zhu, Zhiting Hu, and Eric Xing. 2019. Text infilling. *arXiv preprint arXiv:1901.00158*.

A Appendix

A.1 Few-Shot Learning Details

Choosing appropriate examples for a few-shot prompt can be challenging as task performance is often sensitive to minor changes in prompt design (Zhao et al., 2021). We experimented with prompts randomly selected from the C4, Reddit Writing Prompts, and ROC Stories training sets, as well as prompts consisting of examples handwritten by the authors with the goal of story-writing in mind. For each prompt source, we randomly generated five possible prompts, each with three examples. To simplify the task, we conditioned on desired length but did not include goal conditioning.

An example prompt is shown in Figure A5. When choosing random few-shot prompts from the dataset train sets, in order to keep the few-shot prompt text within the 512-token context length limit of the LLM(Thoppilan et al., 2022) we used for inference, we only considered examples that contained 100 or fewer tokens, so that the max length of the few-shot prompt was no more than 300 tokens. This left 212 tokens for the text of the actual example we were interested in performing the FITB task on. For our hand-written prompt, we wrote the seven examples shown in Table A8. We generated 5 possible prompts by randomly subsampling 3 examples out of these 7 five times.

Table A6 shows the perplexity of the generations from each few-shot prompt. We note that even leaving room for 212 tokens worth of context text, some evaluation examples did not fit in the prompt length, and these examples were skipped when doing this analysis. Figure A4 shows a histogram of the fraction of validation set examples that remained for each few-shot prompt after the too-long examples were filtered out. Based on these results, we chose to include in human evaluation the best few-shot prompt from ROCFILLMIDDLE and the best few-shot prompt from C4FILLBLANK. Figure 2 in the main paper shows the result from the C4FILLBLANK few-shot prompt, whose outputs were rated slightly higher by human annotators.

Our analysis of few-shot learning prompts was not sufficiently exhaustive to rule out the possibility there might exist a prompt for which this technique would be effective. For example, we did not conduct formal experiments to systematically vary the prompt wording/formatting shown in Figure A5. What we can conclude is that the process of finding an ideal prompt requires time-consuming trial-and-error and is quite difficult!

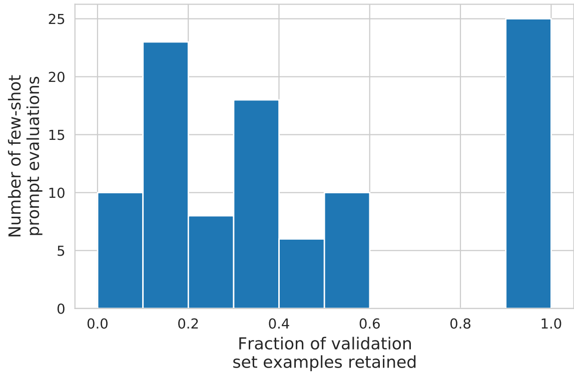

Figure A4: For many of the (validation set, few-shot prompt) combinations, not all validation set examples fit into the maximum sequence length for the LLM. The x-axis on this figure is the fraction of validation set examples which were retained after too-long examples were filtered out. The y-axis is the count of (validation set, few-shot prompt) pairs.

XL Model	Context	Length
C4FILLBLANK	0.867	0.810
RWPFILLBLANK	0.800	0.830
C4FILLEND	0.864	0.826
RWPFILLEND	0.830	0.820
Large Model	Context	Length
C4FILLBLANK	0.860	0.877
RWPFILLBLANK	0.797	0.881
C4FILLEND	0.858	0.775
RWPFILLEND	0.791	0.746

Table A5: Accuracy of models finetuned on FILLBLANKCONT at correctly using provided length and goal conditioning signals.

A.2 Experimenting with Prefix Tuning

During the course of this study, we experimented with the usage of Prefix Tuning (Li and Liang, 2021) for the FITB task. In this method, a fixed-length continuous space prefix is appended to the input sequences and this prefix is directly optimized to maximize performance on a given task. This can be used to estimate an upper bound for the performance of few-shot learning on a given task. We trained two prefixes, both of length 5, on pre-trained GPT-2 of size medium (345M) and large (774M) (Radford et al., 2019). While our results showed that the prefix successfully instructed the pre-trained model to perform the FITB task, neither of these models outperformed our few-shot prompts during human evaluation. In fact, they showed only marginally better performance than our random baseline. Due to the discrepancy in

Prompt

```
Fill in the blank with about 16 words.
Text: "We have to leave now!" Sarah shouted. ____ The
only way out was up. We climbed flight after flight. The
sound of the monsters banging on the door below became
more distant but no less threatening.
Answer: "The zombies are going to break through any
moment, and then we'll all be goners."

Fill in the blank with about 32 words.
Text: I was minding my business at the park, when I was
approached by a little girl who was crying because she
had lost ____ so of course I helped search.
Answer: her cat, which she had just received for her
birthday. She did not want her parents to know she'd al-
ready lost him. I'm a good person

Fill in the blank with about 8 words.
Text: The sun was shining, and little gusts of wind
brought through the window ____ shocking contrast from
the stale city smells she had grown used to.
Answer: the faint scents of honeysuckle and freshly
turned soil. It was a

Fill in the blank with about 8 words.
Lina went to see how candy canes were made. She watched
as the workers added dye to the hot candy. ____ Finally,
they shaped it into a cane and let it cool. Lina felt a
new appreciation for candy canes.
Answer:
```

Target Continuation

```
Then, they stretched it out to make it shiny.
```

Figure A5: In blue, one of the few-shot prompts that was derived from handwritten examples, and in green, the target example we would like to perform infilling on.

parameter count between the prefix tuned GPT-2 models and the LLM model we tested for few-shot prompting, we chose to leave these results out of the final analysis. Future work should seek to explore the limitations of prefix/prompt tuning techniques and the ways in which they and few-shot learning can be fairly compared.

A.3 Finetuning Implementation Details

For length conditioning, when discretizing the target sequence's length to a length bucket, we choose the closest value in $\{1, 2, 4, 8, 16, 32, 64\}$ to the target's length in words.

All training was done in the Mesh Tensorflow T5 codebase.[2] Each T5 model was finetuned for about 50,000 steps with a batch size of 128 examples (i.e., ~6.4M examples were seen during finetuning.) A constant learning rate of 0.0008 was used, and no overfitting was observed.

A.4 Further Finetuning Experiments

In the main paper, we focused on a single finetuning setting, one where half the examples have randomly placed blanks and the other half have blanks always at the end. We actually experimented with three possible finetuning settings:

- In the standard FILLBLANK setting, the blank location is sampled uniform randomly across the sequence.
- In the FILLBLANKCONT setting, for half of the examples the blank is randomly selected and for the other half it is always at the end. As we hypothesized that finetuning on such data would result in better performance at the continuation task, this was the setting we used in the main paper.
- In the CONT (a.k.a. LM-ADAPTION) setting, the blank is always placed at the end of the sequence. In essence, we are finetuning solely for the continuation objective.

For the FILLBLANKCONT setting from the main paper, we additionally experimented with finetuning a 3B parameter "XL" T5 model.

Table A7 shows the perplexity of all these models on a variety of validation sets. Note that these are perplexities in the conventional definition–perplexity of the target sequence given the input sequence using examples from the validation set–not the fluency measure we report in the main paper.

The perplexity numbers across the different models are comparable, since all models used the default T5 vocabulary. The perplexity numbers across different datasets are not comparable since some datasets, like ROC Stories, are simply easier to model than others. Unsurprisingly, the larger models achieved lower perplexity on all validation sets. We can also see from Table A7 that it was probably not strictly necessary to enforce that 50% of training examples had blanks at the end. The model finetuned exclusively with randomly placed blanks (FILLBLANK) performed only slightly worse (probably not statistically significant) on the continuation-style validation sets than the FILLBLANKCONT-trained model.

Finally, Table A5 shows the accuracy of both model sizes on the two conditioning signals which were incorporated: length bucket and goal conditioning. Surprisingly, using a larger model improves goal conditioning accuracy but hurts length conditioning accuracy.

A.5 Further Human Evaluation Details

A screenshot of the Human Intelligence Task (HIT) used for annotations is shown in Figure A6. Workers were paid originally paid $1.85 per HIT, but

[2] https://github.com/google-research/text-to-text-transfer-transformer

Few-shot source:	C4FILL BLANK	ROCFILL MIDDLE	RWPFILL BLANK	RWPFILL BLANK-Sent
C4FILLBLANK	15.67	19.72	**19.65**	**16.82**
ROCFILLMIDDLE	**14.14**	19.61	**19.48**	**16.36**
RWPFILLBLANK	24.39	20.29	32.33	28.13
RWPFILLBLANK-Sent	18.91	**18.21**	24.44	19.87
FS CUSTOM	17.98	19.80	21.72	18.38
Finetuned T5 XL	9.99	19.00	13.64	10.03
Finetuned T5 Large	10.33	20.47	14.08	10.37

Table A6: Perplexity of evaluation sets when the blank has been filled in using LLM with few-shot prompting (top) and our best fine-tuned T5 model (bottom). Among the few-shot results, the best method for each dataset is bolded, as well as methods within one standard error.

Pre-trained model	Finetune setting	C4FILL BLANK	C4FILL END	ROCFILL MIDDLE	ROCFILL END (T)	RWPFILL BLANK	RWPFILL SENTBLANK	RWPFILL END
T5 Large	FILLBLANKCONT	11.79	13.47	6.43	**6.73**	16.15	**14.84**	**19.89**
T5 Large	FILLBLANK	**11.64**	13.88	**6.41**	6.84	**16.11**	14.89	20.16
T5 Large	CONT	16.10	**13.26**	37.08	6.79	21.35	27.73	19.90
T5 XL	FILLBLANKCONT	9.53	11.15	5.34	5.79	13.05	11.98	16.57

Table A7: The perplexity according to T5 Large finetuned with three possible training data settings, with blanks placed randomly (FILLBLANK), with blanks placed always at the end (CONT), or with an equal mix of these two (FILLBLANKCONT). For the large-sized models, the one that achieved lowest perplexity on each dataset is bolded.

since the average HIT duration ended up being 15 minutes, we awarded each rater a bonus to raise their pay to an average of $10 per hour. We restricted the HITs to workers for whom Masters had been granted and who had previously done at least 100 HITs.

Each example was shown to three raters, and annotations were rejected if the rater gave a lower overall score to the random output than to the ground-truth one. A total of 3 annotations were rejected. Overall, the Fleiss' kappa agreement of pairs of annotators giving the same numerical score to the same question was 0.26.

Context	**Target**
An elderly man was sitting alone on a dark path. The man looked down at his feet, and realized ____ . It was a plain pine box and looked as if it had been there for a long time. The man was afraid to look inside the box.	he was holding a bright red box made of pine
The mantle was cluttered with objects: ____ and more than one vase of dried flowers. The bejeweled lamp was at the very back, nearly invisible.	picture frames showing grandchildren and long-ago weddings, knickknacks collected from all over the world,
"We have to leave now!" Sarah shouted. ____ The only way out was up. We climbed flight after flight. The sound of the monsters banging on the door below became more distant but no less threatening.	"The zombies are going to break through any moment, and then we'll all be goners."
The sun was shining, and little gusts of wind brought through the window ____ shocking contrast from the stale city smells she had grown used to.	the faint scents of honeysuckle and freshly turned soil. It was a
I was minding my business at the park, when I was approached by a little girl who was crying because she had lost ____ so of course I helped search.	her cat, which she had just received for her birthday. She did not want her parents to know she'd already lost him. I'm a good person
It was a cold night, and a storm was raging out at sea. A lightning bolt lit up the sky, briefly illuminating the lighthouse ____ plummeted but just before reaching the churning water, he disappeared in a poof of purple flame!	and the young man peering hesitantly over the sheer cliff. Before the next peal of thunder he jumped. At first he
The magician pulled out of his pocket ____ and then a second one and a third. He didn't stop until soon the ground was covered with them.	a scarlet handkerchief

Table A8: Hand-written fill-in-the-blank examples used for "custom" prompt during few-shot learning.

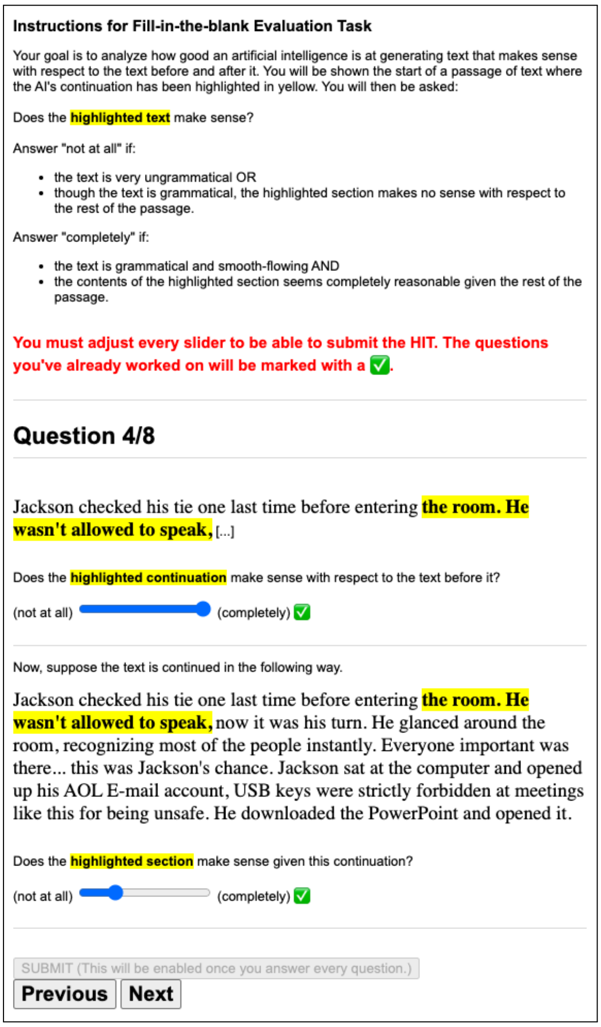

Figure A6: A screenshot of the question structure for human evaluation.