Triggering Multi-Hop Reasoning for Question Answering in Language Models using Soft Prompts and Random Walks

Kanishka Misra★
Purdue University
kmisra@purdue.edu

Cicero Nogueira dos Santos
Google Research
cicerons@google.com

Siamak Shakeri
Google DeepMind
siamaks@google.com

Abstract

Despite readily memorizing world knowledge about entities, pre-trained language models (LMs) struggle to compose together two or more facts to perform multi-hop reasoning in question-answering tasks. In this work, we propose techniques that improve upon this limitation by relying on random walks over structured knowledge graphs. Specifically, we use soft prompts to guide LMs to chain together their encoded knowledge by learning to map multi-hop questions to random walk paths that lead to the answer. Applying our methods on two T5 LMs shows substantial improvements over standard tuning approaches in answering questions that require 2-hop reasoning.

1 Introduction

Performing multi-hop reasoning to answer questions such as *Where was David Beckham's daughter born?* requires two fundamental capacities: **C1:** possessing pre-requisite knowledge (*David Beckham's daughter is Harper Beckham*, *Harper Beckham was born in Los Angeles*), and **C2:** ability to compose internalized knowledge. Contemporary pre-trained language models (LMs) such as BERT (Devlin et al., 2019) and T5 (Raffel et al., 2020) have been shown to be adept at encoding factual knowledge (Petroni et al., 2019; Zhong et al., 2021; Roberts et al., 2020), an ability that can be further boosted by explicitly integrating them with knowledge about entities and relations (Bosselut et al., 2019; Sun et al., 2020; Wang et al., 2021, *i.a.*). At the same time, these LMs often struggle to compose the knowledge they encode (Kassner et al., 2020; Talmor et al., 2020; Moiseev et al., 2022), and therefore do not satisfy **C2**. To overcome this limitation, previous works have proposed methods that decompose multi-hop questions into single hop sub-questions that models can more easily answer (Min et al., 2019; Perez et al., 2020, *i.a.*). However, such methods require training entirely separate models, or make use of human-annotations (Patel et al., 2022). Furthermore, they focus on tasks where models explicitly receive additional text containing relevant facts, which makes it unclear if they can *truly* compose the knowledge that they have internalized.

In this work, we aim to improve the standalone, self-contained ability of LMs to perform multi-hop reasoning. We posit that *random walks*—paths between entity nodes sampled from structured knowledge graphs—can provide a useful training signal for LMs to compose entity knowledge. To test this, we perform a case-study on two T5 models (LARGE and XXL, Raffel et al., 2020). Specifically, we first integrate within the LMs the single-hop knowledge that is required to answer multi-hop questions (effectively guaranteeing **C1** is met). We show that this alone is not enough to demonstrate substantial improvements on questions requiring 2-hop reasoning. We then adapt the knowledge integrated T5 models by training soft prompts (Qin and Eisner, 2021; Lester et al., 2021) on random walks over the structured knowledge that they have encoded, and devise two methods that trigger this ability in the LMs given a multi-hop question as input. The first method, **Parse-then-Hop** (PATH), uses two specialized soft prompts: one to parse entities and relations from the question, and another to generate a path to the answer, resembling the outputs of a random walk. The second method, **MIXHOP**, trains a single prompt on a mixture that combines the QA task with the random walk training, so as to allow the model to implicitly learn PATH's task. Both these soft prompt methods use the same underlying LM (kept frozen), and guide it to compose its internalized entity knowledge.

Our experiments suggest that integrating random walks in the T5 models using our proposed techniques can substantially improve their ability to

★ Work done during an internship at Google Research.

Findings of the Association for Computational Linguistics: ACL 2023, pages 972–985
July 9-14, 2023 ©2023 Association for Computational Linguistics

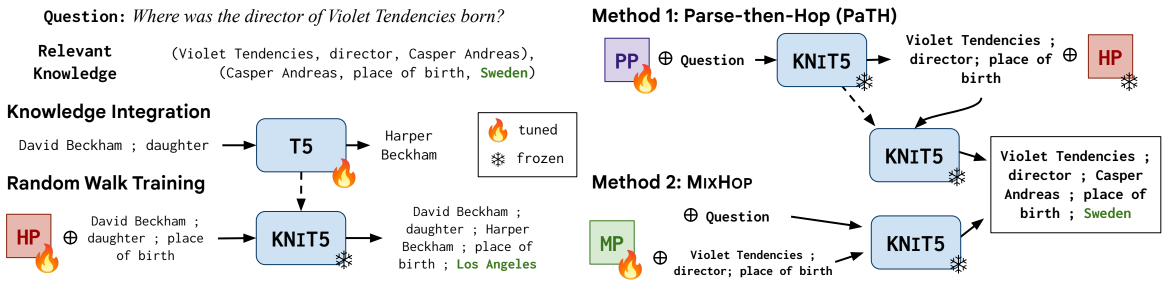
Question: Where was the director of Violet Tendencies born?
Relevant Knowledge
(Violet Tendencies, director, Casper Andreas),
(Casper Andreas, place of birth, Sweden)
Knowledge Integration
David Beckham ; daughter
T5
Harper Beckham
tuned
frozen
Random Walk Training
HP
⊕
David Beckham ; daughter ; place of birth
KNIT5
David Beckham ; daughter ; Harper Beckham ; place of birth ; Los Angeles
Method 1: Parse-then-Hop (PaTH)
PP
⊕ Question
KNIT5
Violet Tendencies ; director; place of birth
⊕
HP
KNIT5
Violet Tendencies ; director ; Casper Andreas ; place of birth ; Sweden
Method 2: MIXHOP
⊕ Question
MP
⊕ Violet Tendencies ; director; place of birth
KNIT5

Figure 1: Overview of our approach. Colored rectangular boxes indicate soft prompts: Hopping Prompts (HP), Parsing Prompts (PP), and Prompts for the MIXHOP approach (MP). ⊕ indicates concatenation.

answer entity-centric 2-hop questions (Ho et al., 2020) at larger model sizes. Briefly, on T5-XXL our methods show improvements over previously proposed prompt-tuning approaches (Lester et al., 2021; Vu et al., 2022) as well as full model fine-tuning, with PATH and MIXHOP demonstrating gains of ∼16 and ∼9.6 points in exact match scores over fine-tuning the entire model, respectively. In the case of T5-LARGE, our methods demonstrate improvements over standard prompt-tuning methods, but fall short of the performance achieved using fine-tuning, suggesting that larger models—with up to 11B parameters—are more conducive to leveraging the training signal provided by random walks via soft prompts.

2 Method

2.1 Models

We apply our methods on two T5.1.1 models (Raffel et al., 2020)—T5-LARGE (770M parameters) and T5-XXL (11B parameters), using checkpoints that have been adapted using the Prefix LM objective for 100K steps (Lester et al., 2021).

2.2 Knowledge Integration

We first ensure that the LMs we use have the prerequisite single-hop knowledge (C1) required to answer multi-hop questions. This is necessary, as preliminary experiments suggested that the T5 models we used did not satisfy this primary criterion for multi-hop reasoning (see Table 1). Specifically, we follow Bosselut et al. (2019) and fine-tune our LMs on knowledge graph (KG) triples containing the relevant knowledge that is to be composed to answer questions. That is, given a triple (e_1, r, e_2), where e_1 and e_2 are entities, and r is the relation, we fine-tune our T5 models to take as input the string “`e1 ; r1`”, and produce “`e2`” as output, using the Prefix LM objective (Raffel et al., 2020). To avoid catastrophic forgetting (McCloskey and Cohen, 1989) and retain the LMs’ language understanding abilities, we mix our knowledge integration training instances with that of the models’ pre-training corpus—i.e., C4 (Raffel et al., 2020)—in a 50:50 mixture. We denote the resulting models as **KNowledge-Integrated T5** (KNIT5).

2.3 Composing knowledge using soft prompts

Random Walk training Our method is centered around guiding the KNIT5 LMs to chain together their encoded knowledge by training them on random walks over a relevant KG. We formulate random walks here as as a sequence of entity-relation-entity triples that are connected linearly via shared entities. Figure 1 shows an example with a random walk of length 3 (`Violet Tendencies ; director ; Casper Andreas ; place of birth ; Sweden`). To perform our random walk training, we rely on soft prompts (Li and Liang, 2021; Lester et al., 2021; Qin and Eisner, 2021), a sequence of learnable token-vectors that are prepended to the input of the LM. Importantly, we only update these vectors during training, thereby keeping intact the utility and encoded knowledge of the main LM, while also being parameter efficient. Our training procedure is as follows: we first perform uniform random walks of length n over the KG used in section 2.2, resulting in a set whose elements are sequences of entities interleaved by the relations that connect them: $(e_1, r_1, e_2, \ldots, r_{n-1}, e_n)$. During training, KNIT5 receives as input an incomplete path, with only the initial entity and the intermediate relations $(e_1, r_1, r_2, \ldots, r_{n-1})$, and is tasked to generate the full path: $(e_1, r_1, e_2, r_2 \ldots, r_{n-1}, e_n)$. We denote the trained prompts that trigger this ability in KNIT5 as **Hopping Prompts**.

2.4 Performing QA using Hopping Prompts

We propose two new techniques that utilize Hopping Prompts to map natural language questions to appropriate paths in the knowledge graph:

Parse-then-Hop (PATH) We take advantage of the modularity of soft prompts, and distribute the responsibility of parsing the relational structure from questions and random walk querying using separate specialized prompts, keeping the underlying model the same. We train "parsing" prompts that parse questions to incomplete random walk queries, resembling the inputs to the Hopping Prompts described above. For instance, the question "*Where was David Beckham's daughter born?*" is parsed to "`David Beckham ; daughter ; place of birth`". We then swap the parsing prompts with the hopping prompts, using the outputs from the parsing step as inputs and then run inference to get a path from the entity in the question to the answer: "`David Beckham ; daughter ; Harper Beckham ; place of birth ;` **`Los Angeles`**", as shown in Figure 1. We posit that parsing of the appropriate relational structure from the question should be easy and self-contained, since it only involves using the surface form of the question as opposed to invoking any external knowledge, which is delegated to Hopping Prompts.

MIXHOP We propose to jointly train a single set of prompts on a mixture of the QA task and the Hopping Prompts task (50:50), thereby halving the number of forward passes from the previous method. Our primary motivation here is to provide diverse training signals that get models to map questions to the structured knowledge that explicitly connects the entity in the question to the answer entity. Like PATH, MIXHOP directly produces random walk paths as output, as shown in Figure 1.

3 Experimental Setup

3.1 Data

Multi-hop QA Dataset While traditional multi-hop QA datasets provide additional paragraphs (Yang et al., 2018; Trivedi et al., 2022) for models to reason over, we operate under the more challenging closed-book QA setting (Roberts et al., 2020), where such contexts are omitted. Specifically, we use the "compositional" and "inference" subsets of the **2WikiMultiHopQA** dataset (Ho et al., 2020), which contains 2-hop English questions focusing on 98,284 entities and 29 relations, sourced from WikiData (Vrandečić and Krötzsch, 2014). We select this dataset as it uniquely provides the *precise* structured knowledge that is required to answer each question, in the form of entity-relation-entity triples.[1] Since the test splits for these specific subsets are private, we use the validation split as the test set, and use 10% of the training set for validation. In total we have 72,759 train, 8,085 validation, and 6,768 test questions.

1-hop QA Dataset To characterize if the models we test have the pre-requisite 1-hop knowledge, we additionally construct 1-hop questions from 2WikiMultiHopQA by applying manually defined templates over the entity triples provided for each 2-hop question (see Appendix C). For instance, the triple `Inception ; director ; Christopher Nolan` is converted to *Who is the director of Inception?*. We end up with 83,643 train, 5,022 validation, and 6,440 test QA instances. We term this constructed dataset as **1WikiHopQA**.

Knowledge Integration Data We build the KG for our methods using the set of ground-truth triples provided in the 2WikiMultiHopQA dataset (98,284 entities and 29 relations, amounting to 95K triples).

Random Walk Training Corpus For each entity in the above KG, we sample *up to* 20 random walks of length 3, each corresponding to an instance of 2 hops between entities. We repeat this step 5 times with different seeds, discard duplicate paths, and end up with a total of 165,324 unique paths as a result. **Importantly, we hold out the paths that include the triples in the QA task's validation and test sets in order to avoid leakage**, ending up with 155,311/ 8,085/6,768 paths as our train/validation/test sets, respectively. This way, our experiments test for the kinds of generalization where models should successfully place entities in novel structures (complete paths in the KG), whose primitive knowledge (1-hop triples) is encoded in the model, but the composition is not. This can be viewed as a partial version of the lexical and structural generalization tests in stricter, more prominent compositional generalization benchmarks (Lake and Baroni, 2018; Kim and Linzen, 2020).

3.2 Baselines and Comparisons

We compare our proposed approaches to standard fine-tuning and prompt-tuning (Lester et al., 2021),

[1] Works such as Balachandran et al. (2021) propose unsupervised mappings of questions in more popular datasets such as NaturalQuestions (Kwiatkowski et al., 2019) to paths in knowledge graphs, but our initial investigations of these paths found them to be extensively noisy.

Setup	Model	LARGE	XXL
PT	T5	4.36	6.89
	KnIT5	**6.30**	**31.64**
FT	T5	6.24	8.82
	KnIT5	**22.73**	**43.60**

Table 1: Test EM scores achieved by T5 and KnIT5 on 1WikiHopQA. PT: Prompt-Tuning, FT: Fine-Tuning.

Model	EM	F1
KnIT5-LARGE	22.83	84.72
KnIT5-XXL	**58.36**	**92.82**

Table 2: Best reported validation EM and F1 scores achieved from training Hopping Prompts to get KnIT5 models to generate random-walks. $N = 8085$.

which we use to directly produce the answer, without any intermediate entities or relations. Additionally, we also adapt SPoT (Vu et al., 2022), a prompt-tuning method where we initialize prompts with those that were pre-trained on related tasks. In our adaptation, we initialize prompts using the values of the Hopping Prompts, and SPoT-transfer them to guide KnIT5 models to generate the full output, similar to PATH and MixHop. Since we operate in the closed book QA setting (Roberts et al., 2020), our methods cannot be directly compared to previous approaches on the dataset we considered, all of which receive paragraph contexts during training. Only two other methods have considered the present dataset in its closed-book format (Press et al., 2023; Wang et al., 2022). However, both of them use smaller subsets of the validation set as their testing set, and test on different pretrained models, making it impractical to directly compare our results to their reported values.

4 Experiments and Findings[2]

We report and summarize our results as follows:

Integration of 1-hop knowledge only results in marginal improvements on 2-hop questions We begin by first establishing the extent to which T5 models encode and compose 1-hop knowledge required to answer 2-hop questions, and whether additional knowledge integration (via KnIT5) can improve both these abilities. From Tables 1 and 3, we observe that the T5 models struggle to answer both 1-hop as well as 2-hop questions, suggesting that they critically lack the precise 1-hop entity knowledge required to demonstrate success on the 2-hop questions. The KnIT5 LMs overcome this limitation, by showing substantial gains on 1WikiHopQA over their T5 counterparts—they show improvements of $\sim$16.5 and $\sim$34.8 points in exact match (EM) scores at LARGE and XXL sizes in the fine-tuning setting, respectively (Table 1). However, this is insufficient to show improvements on 2-hop questions—where maximum gain over T5 is only 2.2 points, achieved by prompt-tuning KnIT5-XXL (see Table 3). This suggests that even after being endowed with the prerequisite 1-hop knowledge, both LMs are unable to successfully answer more complicated questions, echoing the results of Moiseev et al. (2022). Note that both KnIT5 models almost perfectly memorize the KG in our knowledge-integration experiments (achieving $\sim$96% EM in under 10K training steps; see Appendix B.1), so their limitations on 2-hop questions are likely not due to lack of entity knowledge and perhaps instead due to the inability to compose or chain together memorized facts.

Generalizing to novel random walks may require the prompt-tuning of larger LMs We now turn to analyzing the performance of models in generating random walks, a critical component for all our proposed QA methods. How well does prompt-tuning LMs generalize to KG paths composed of facts they have memorized but are unseen during training? Recall that this step involved leveraging soft prompts (called Hopping Prompts) to guide the LMs to chain together their memorized entity knowledge and generate paths akin to performing a random walk. That is, it is the Hopping Prompts that must provide the necessary condition in the encoder to facilitate successful output-generation, and not the entire LM. Also recall that we explicitly held out the paths involving triples in the validation and test sets of the main QA task to prevent complete memorization (due to leakage into the training set). This way we are able to measure the extent to which models learned to construct KG paths in a generalized manner. To this end, we compute the EM and F1 scores over the full generated spans of entities, interleaved by the relations that connect them. Note that EM is substantially stricter than F1, since F1 rewards par-

[2] Training details for all experiments can be found in Appendix A.

Size	Prompt-Tuning		Fine-Tuning		SPoT	PATH	MIXHOP
	T5	KNIT5	T5	KNIT5			
LARGE	4.47	5.29	10.03	**11.19**	7.22	8.62	6.58
XXL	6.42	8.62	12.92	13.47	20.03	**29.37**	23.09

Table 3: Test set EM scores achieved by various tuning methods on 2WikiMultiHopQA (Ho et al., 2020). SPoT (Vu et al., 2022), PATH, and MIXHOP use KNIT5 as their base model.

tial overlap of tokens between the target vs. the generated output. Table 2 shows these scores for KNIT5-LARGE and KNIT5-XXL on the validation set of our random walk task, tuned using the Hopping Prompts. We see from Table 2 that there is a substantial gap between KNIT5-LARGE (∼23 EM) and KNIT5-XXL (∼58 EM), suggesting that the LARGE model finds it difficult to generalize to random walk paths involving entities and relations outside of the training set. We conclude from this observation that the gap between KNIT5-LARGE and KNIT5-XXL in generalizing to held-out KG paths is likely going to be reflected when tested for 2-hop QA. That is, we expect our prompting methods with KNIT5-LARGE as the base-model to struggle on our test set questions as their ground-truth paths were not encountered during training, and at the same time, expect the opposite to be the case for KNIT5-XXL. Additionally, the EM score achieved by the XXL-sized model is well below perfect values, highlighting important avenues for future work to improve upon these gaps.

Training on random walks substantially improves 2-hop capabilities ..but mostly in larger LMs We used three methods that leveraged the training signal provided by random walks to compose the 1-hop knowledge as memorized by KNIT5: PATH (ours), MIXHOP (ours), and SPoT (Vu et al., 2022). Due to lack of space, examples of the outputs from each of these methods, along with analysis of intermediate steps (e.g., parsing) are shown in Appendix B. We observe from Table 3 that for the XXL-sized model, all three methods lead to substantial improvements in performance on 2-hop questions over standard tuning approaches on T5 and KNIT5. Notably for KNIT5-XXL, random walk-integrated methods improve even over fine-tuning, which is often expected to be better at transfer learning as compared to parameter efficient methods. Among the three, our PATH method shows the best improvements (∼16 point gain over fine-tuning KNIT5-XXL) at answering 2-hop questions. This showcases the promise of learning separate specialized prompts that operate over the same underlying model to first parse natural language into incomplete structured knowledge, and then expand it to answer the question, while also eliciting intermediate steps (Wang et al., 2022), similar to recent in-context prompting methods (Wei et al., 2022b; Nye et al., 2022). While the MIXHOP method (∼9.6 point gain over fine-tuning) falls short of PATH, it still improves over SPoT (∼6.6 point gain over fine-tuning), suggesting that joint training of related tasks may improve over sequential training (as employed by SPoT) in performing multi-hop reasoning, at larger model sizes. In the case of T5-LARGE and KNIT5-LARGE, while the proposed methods show improvements over standard prompt-tuning, with PATH demonstrating a gain of 3.33 points over prompt-tuning KNIT5-LARGE, they fall-short of the performance achieved by fine-tuning. However, their non-trivial improvements over regular prompt-tuning suggests the general benefits of the training signal provided by random walks, which end up being most impressive at models that are an order of magnitude larger. Overall, these results corroborate with our hypothesis from the random walk tests about KNIT5-LARGE's potential inability to generate partially novel random walks given either natural language multi-hop questions (MIXHOP) or their parses (PATH).

5 Conclusion

We show that composition of memorized world knowledge can be triggered in LMs with up to 11B parameters (T5-XXL) to a desirable extent by leveraging training signal from random walks over structured knowledge using approaches based on prompt-tuning (Lester et al., 2021). Doing so leads to substantial improvements in the LMs' ability to answer 2-hop questions, even beyond standard, full model fine-tuning.

Limitations

Despite showing non-trivial improvements in the multi-hop capabilities of T5 models, our work has multiple limitations.

Restricted to 2-hops First, we chose 2WikiHopMultiQA (Ho et al., 2020) as our primary dataset since it uniquely maps each question to a chain of triples that contain the precise, noiseless single-hop knowledge required to answer the question. However, this comes at the cost of our analyses only being restricted to 2-hops (though see arguments by Press et al. (2023, sec 3.5) who suggest 3-and-4-hop questions to be too convoluted to understand even by native-speakers). Nonetheless, our random walk training method is general by definition, and can be extended to multiple hops, though its effectiveness on QA tasks requiring more than 2-hops of reasoning remains to be measured.

Knowledge Graph size Our focus in this paper was to allow models to chain together their internalized knowledge in order to answer complex 2-hop questions. However, this critically requires them to possess the world knowledge required to answer the questions, for which we had to memorize the KG constructed using the structured triples provided in the dataset. This trade-off between focusing on knowledge composition vs. fully encoding world knowledge restricted our KG to be small in size (only 98,284 entities and 29 relations), which could be impractical in most real-world applications. In future work, we will experiment with larger sized KGs (Vrandečić and Krötzsch, 2014), by adding a substantially larger amount of additional triples to the existing KG, and measure their impact on multi-hop reasoning.

Lack of diverse QA tasks Finally, we were unable to consider popular datasets with CBQA versions such as TriviaQA (Roberts et al., 2020), NaturalQuestions (Kwiatkowski et al., 2019), etc., due to their lack of links from questions to structured knowledge. Future work can apply entity and relational linking techniques (Balachandran et al., 2021; Agarwal et al., 2021) in order to augment such QA datasets with (possibly) noisy links to structured knowledge, which will allow us to paint a more holistic picture of our methods. Additionally, this would also overcome the above limitation (of KG size), as it would substantially increase the amounts of entities and relations to be encoded within models.

Implications for Larger Models Although we show clear improvements in triggering 2-hop reasoning in the largest T5 LM (T5-XXL), with 11B parameters, contemporary work has shown that multi-step reasoning capacities naturally emerge in LMs that are two or three orders of magnitude larger (Brown et al., 2020; Chowdhery et al., 2022; Wei et al., 2022b,a). However, these LMs benefit from examples in-context (especially since tuning them is non-trivial and expensive), and therefore it is unclear whether our methods can improve such models' capacities even further. We have not tested such LMs in our work, due to resource limitations.

Acknowledgments

We thank Noah Constant, Chung-Ching Chang, Brian Lester, and Ben Withbroe from Google Research for their helpful comments and advice. We would also like to thank our three anonymous reviewers for their useful feedback.

References

Oshin Agarwal, Heming Ge, Siamak Shakeri, and Rami Al-Rfou. 2021. Knowledge graph based synthetic corpus generation for knowledge-enhanced language model pre-training. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3554–3565, Online. Association for Computational Linguistics.

Vidhisha Balachandran, Bhuwan Dhingra, Haitian Sun, Michael Collins, and William Cohen. 2021. Investigating the effect of background knowledge on natural questions. In *Proceedings of Deep Learning Inside Out (DeeLIO): The 2nd Workshop on Knowledge Extraction and Integration for Deep Learning Architectures*, pages 25–30, Online. Association for Computational Linguistics.

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.

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

Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, et al. 2022. Palm: Scaling language modeling with pathways. *arXiv preprint arXiv:2204.02311*.

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.

Xanh Ho, Anh-Khoa Duong Nguyen, Saku Sugawara, and Akiko Aizawa. 2020. Constructing a multi-hop QA dataset for comprehensive evaluation of reasoning steps. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 6609–6625, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Nora Kassner, Benno Krojer, and Hinrich Schütze. 2020. Are pretrained language models symbolic reasoners over knowledge? In *Proceedings of the 24th Conference on Computational Natural Language Learning*, pages 552–564, Online. Association for Computational Linguistics.

Najoung Kim and Tal Linzen. 2020. COGS: A compositional generalization challenge based on semantic interpretation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 9087–9105, Online. Association for Computational Linguistics.

Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, Kristina Toutanova, Llion Jones, Matthew Kelcey, Ming-Wei Chang, Andrew M. Dai, Jakob Uszkoreit, Quoc Le, and Slav Petrov. 2019. Natural questions: A benchmark for question answering research. *Transactions of the Association for Computational Linguistics*, 7:452–466.

Brenden Lake and Marco Baroni. 2018. Generalization without Systematicity: On the Compositional Skills of Sequence-to-Sequence Recurrent Networks. In *International conference on machine learning*, pages 2873–2882. PMLR.

Brian Lester, Rami Al-Rfou, and Noah Constant. 2021. The power of scale for parameter-efficient prompt tuning. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 3045–3059, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

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, Online. Association for Computational Linguistics.

Michael McCloskey and Neal J Cohen. 1989. Catastrophic interference in connectionist networks: The sequential learning problem. In *Psychology of learning and motivation*, volume 24, pages 109–165. Elsevier.

Sewon Min, Victor Zhong, Luke Zettlemoyer, and Hannaneh Hajishirzi. 2019. Multi-hop reading comprehension through question decomposition and rescoring. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 6097–6109, Florence, Italy. Association for Computational Linguistics.

Fedor Moiseev, Zhe Dong, Enrique Alfonseca, and Martin Jaggi. 2022. SKILL: Structured knowledge infusion for large language models. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1581–1588, Seattle, United States. Association for Computational Linguistics.

Maxwell Nye, Anders Johan Andreassen, Guy Gur-Ari, Henryk Michalewski, Jacob Austin, David Bieber, David Dohan, Aitor Lewkowycz, Maarten Bosma, David Luan, et al. 2022. Show your work: Scratchpads for intermediate computation with language models. In *Deep Learning for Code Workshop*.

Pruthvi Patel, Swaroop Mishra, Mihir Parmar, and Chitta Baral. 2022. Is a question decomposition unit all we need? In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 4553–4569, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Ethan Perez, Patrick Lewis, Wen-tau Yih, Kyunghyun Cho, and Douwe Kiela. 2020. Unsupervised question decomposition for question answering. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 8864–8880, Online. Association for Computational Linguistics.

Fabio Petroni, Tim Rocktäschel, Sebastian Riedel, Patrick Lewis, Anton Bakhtin, Yuxiang Wu, and Alexander Miller. 2019. Language models as knowledge bases? 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 2463–2473, Hong Kong, China. Association for Computational Linguistics.

Ofir Press, Muru Zhang, Sewon Min, Ludwig Schmidt, Noah A. Smith, and Mike Lewis. 2023. Measuring and narrowing the compositionality gap in language models. *ICLR 2023 Submission*.

Guanghui Qin and Jason Eisner. 2021. Learning how to ask: Querying LMs with mixtures of soft prompts. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 5203–5212, Online. Association for Computational Linguistics.

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

Adam Roberts, Colin Raffel, and Noam Shazeer. 2020. How much knowledge can you pack into the parameters of a language model? In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 5418–5426, Online. Association for Computational Linguistics.

Tianxiang Sun, Yunfan Shao, Xipeng Qiu, Qipeng Guo, Yaru Hu, Xuanjing Huang, and Zheng Zhang. 2020. CoLAKE: Contextualized language and knowledge embedding. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 3660–3670, Barcelona, Spain (Online). International Committee on Computational Linguistics.

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.

Harsh Trivedi, Niranjan Balasubramanian, Tushar Khot, and Ashish Sabharwal. 2022. MuSiQue: Multihop questions via single-hop question composition. *Transactions of the Association for Computational Linguistics*, 10:539–554.

Denny Vrandečić and Markus Krötzsch. 2014. Wikidata: A free collaborative knowledgebase. *Communications of the ACM*, 57(10):78–85.

Tu Vu, Brian Lester, Noah Constant, Rami Al-Rfou', and Daniel Cer. 2022. SPoT: Better frozen model adaptation through soft prompt transfer. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5039–5059, Dublin, Ireland. Association for Computational Linguistics.

Boshi Wang, Xiang Deng, and Huan Sun. 2022. Iteratively prompt pre-trained language models for chain of thought. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 2714–2730, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Xiaozhi Wang, Tianyu Gao, Zhaocheng Zhu, Zhengyan Zhang, Zhiyuan Liu, Juanzi Li, and Jian Tang. 2021. KEPLER: A unified model for knowledge embedding and pre-trained language representation. *Transactions of the Association for Computational Linguistics*, 9:176–194.

Jason Wei, Yi Tay, Rishi Bommasani, Colin Raffel, Barret Zoph, Sebastian Borgeaud, Dani Yogatama, Maarten Bosma, Denny Zhou, Donald Metzler, Ed H. Chi, Tatsunori Hashimoto, Oriol Vinyals, Percy Liang, Jeff Dean, and William Fedus. 2022a. Emergent abilities of large language models. *Transactions on Machine Learning Research*.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, brian ichter, Fei Xia, Ed H. Chi, Quoc V Le, and Denny Zhou. 2022b. Chain of thought prompting elicits reasoning in large language models. In *Advances in Neural Information Processing Systems*.

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William Cohen, Ruslan Salakhutdinov, and Christopher D. Manning. 2018. HotpotQA: A dataset for diverse, explainable multi-hop question answering. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2369–2380, Brussels, Belgium. Association for Computational Linguistics.

Zexuan Zhong, Dan Friedman, and Danqi Chen. 2021. Factual probing is [MASK]: Learning vs. learning to recall. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 5017–5033, Online. Association for Computational Linguistics.

A Training and Experiment Details

Hyperparameters We use the default hyperparameters and optimizers used to train the T5 1.1 checkpoints (Raffel et al., 2020) as well as those used in the Prompt-Tuning and SPoT papers (Lester et al., 2021; Vu et al., 2022). We set the prompt-length to 100 for all prompt-tuning experiments, and initialized them with the top 100 tokens in the T5 models' vocabulary, following Lester et al. (2021). We fine-tune and prompt-tune our models for a maximum of 100K and 200K steps, respectively. We stop training on convergence, and use the checkpoint with the best validation performance to evaluate. Tables 4, 5, and 6 show hyperparameter values for each type of experiment. All results are from single runs.

Hardware and Compute Prompt-tuning and fine-tuning experiments for LARGE models were run on 16 TPUv3 chips, while those for XXL models were run on 64 TPUv3 chips. One exception is knowledge integration (which also involved continual pre-training on C4, larger batch size, and longer sequences), for which we used 256 TPUv3 chips for XXL, and 64 TPUv3 chips for LARGE.

Code For metric calculation and checkpoints, we use the T5 and T5x code-base, open-sourced on github.[3][4] For prompt-tuning experiments, we adapt the original code-base (Lester et al., 2021), which is also open-sourced.[5]

Data The 2WikiMultiHopQA dataset (Ho et al., 2020) has been released with Apache 2.0 license.[6]

Hyperparameter	Values
Batch Size	32 (XXL), 128 (LARGE)
Learning Rate	0.001
Dropout	0.1
Training Steps	100K (w/ early stopping)

Table 4: Hyperparameters used for fine-tuning T5-LARGE and T5-XXL. Values except batch size and training steps kept same as Raffel et al. (2020).

Hyperparameter	Values
Batch Size	512
Learning Rate	0.001
Dropout	0.1
Training Steps	100K (w/ early stopping)

Table 5: Hyperparameters used for Knowledge Integration experiments. Values except batch size and training steps kept same as Raffel et al. (2020).

Hyperparameter	Values
Batch Size	32 (XXL), 128 (LARGE)
Learning Rate	0.3
Prompt Length	100
Dropout	0.1
Training Steps	200K (w/ early stopping)

Table 6: Hyperparameters used for all prompt-tuning experiments. Values except batch size kept same as Lester et al. (2021), number of training steps kept same as Vu et al. (2022), who found longer training to be beneficial.

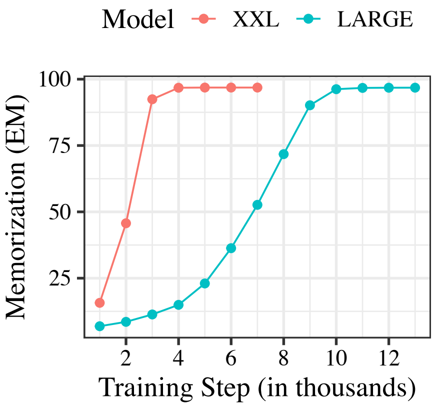

Figure 2: Time course of KG memorization for different KNIT5 model sizes. EM scores calculated for producing object entity (e_2), given subject (e_1) and relation (r) as inputs to T5 models.

B Additional Analyses

B.1 Knowledge Integration

Integrating single-hop entity knowledge is an important part of our methods. How well are the models able to actually encode this knowledge? Figure 2 shows the dynamics of memorization across both models, measured as the exact match scores in generating e_2 given e_1 and r. From Figure 2, we see that the XXL and LARGE models can memorize 96% of the KG within 5,000 and 10,000 steps respectively. With a batch size of 512, this translates to traversing the dataset 27 and 54 times, respectively, for XXL and LARGE. An important caveat here is that the models are also being tuned on C4 (Raffel et al., 2020), in order to retain the models' general language understanding-like capabilities. That is, they can be expected to memorize the KG relatively faster in the absence of training on the C4 corpus, but this would constitute a trade-off, by leading to overfitted models with substantial loss their original utility on other NLP tasks.

B.2 Parsing Step in PATH

The parsing step is essential for our Parse-then-Hop approach to succeed. Here we perform additional analyses on how well models can successfully extract the relational structure that is required to answer the 2-hop questions in 2WikiMultiHopQA. Recall that the objective of the parsing step is to produce as output a sequence indicating an incomplete random walk, containing only the initial entity (seed node), followed by the relations (edges) that

[3] https://github.com/google-research/text-to-text-transfer-transformer/tree/main/t5

[4] https://github.com/google-research/t5x

[5] https://github.com/google-research/prompt-tuning

[6] https://github.com/Alab-NII/2wikimultihop

Model	Relation EM	Entity EM	Full EM
KNIT5-LARGE	98.69	76.19	78.98
KNIT5-XXL	99.17	78.46	80.17

Table 7: Metrics for the parsing sub-task of PATH on test-set questions.

lead to the final entity. For instance, if the question is "*Where was the director of Inception (film) born?*" the output of the parsing step should be:

```
Inception (film) ; director ;
place of birth
```

Here, `Inception (film)` is the entity, e_1, while `director` and `place of birth` are the relations, r_1 and r_2, respectively. We analyze the extent to which models successfully extract these three elements for the 6,768 test set questions, by measuring three quantities: (1) **Relation EM**, which is the exact match score computed between the ground truth span of relation pairs (here "`director ; place of birth`"), and that extracted from the model outputs; (2) **Entity EM**, which is similar to Relation EM, but only considers the initial entity; and (3) **Full EM**, which computes the exact match score between the full output and the target. Table 7 shows these values from prompt-tuning the two KNIT5 models.

From Table 7, we see that prompt-tuning both models allows them to achieve almost perfect EM values in extracting the relation pairs from the questions. However, we notice that models are not able to maintain this performance in copying over the entity, which lowers their overall EM scores on this task. We performed a manual analysis of 50 randomly sampled outputs—with incorrect entity predictions—and found most errors to be due to omission of tokens involving middle names, or additional information about the entity such as the "`(film)`" in the above example (other examples include the entity's title, such as "`Count of East Frisia`", or "`(born in year XXX)`", "`(died in year XXX)`", etc.)

B.3 Example Outputs

Tables 8, 9, 10, and 11 show examples of outputs from the different approaches used in this work (examples shown for the XXL-sized models). Below we discuss each of these cases in detail:

- In Table 8, all approaches that leverage the training signal from random walks succeed, while tuning methods that do not fail. Additionally, all three random walk-integrated methods agree on their parsed relational structure as well as the intermediate entity.

- In Table 9, only the two proposed methods (PATH and MIXHOP) succeed, while all other methods fail. Note that SPOT correctly predicts the correct intermediate entity (`Sally Hemings`), but is unable to predict the final entity (`John Wayles`).

- Table 10 shows an example where all approaches fail. However, this question is ambiguous, as *aunt* can either mean *father's sister* or *mother's sister* – our random walk integrated methods correctly predict these relational structures but are unable to resolve the intermediate and final entities.

- Table 11 shows an example where all approaches are supposedly scored as incorrect, but are in-fact correct. Here we argue that the ground truth answer, "*United Kingdom*" is in its incorrect form, since the question asks for the nationality of a person. Our random walk-integrated methods successfully predict the relational structure and intermediate entities. Moreover all approaches predict `British` or `English`, which are more acceptable forms of nationality for persons from the United Kingdom. This problem could be mitigated by adding in aliases for the entities in the ground-truth answer space, similar to TriviaQA (Roberts et al., 2020).

C Templates for constructing 1WikiHopQA

Here we describe our process of constructing 1WikiHopQA: a collection of English question-answer pairs that only require single-hop knowledge using the 2WikiMultiHopQA (Ho et al., 2020) dataset. The 2WikiMultiHopQA dataset provides unique sequences of single-hop triples that collectively answer each 2-hop question. These amount to a total of 95,103 unique triples spanning 98,284 unique entities and 29 relations. We manually define a diverse set of templates for each relation, as shown in Table 12. For many relations, we have multiple different paraphrases of the question template, e.g., the relation `director` translates to: *Who is the director of X?* or *Who directed the film X?* In such

Question: *Where was the place of burial of the director of film New World (1995 Film)?* **Answer:** Père Lachaise Cemetery

Model	Setup	Response
T5-XXL	FT	**Forest Lawn Memorial Park**
	PT	**Forest Lawn Memorial Park**
KNIT5-XXL	FT	**New York**
	PT	**Forest Lawn Memorial Park**
	SPoT	New World ; director ; Alain Corneau ; place of burial ; **Père Lachaise Cemetery**
	PATH	**PP:** New World ; director ; place of burial
		HP: New World ; director ; Alain Corneau ; place of burial ; **Père Lachaise Cemetery**
	MIXHOP	New World ; director ; Alain Corneau ; place of burial ; **Père Lachaise Cemetery**

Table 8: An example case where methods that leverage random walks succeed, but baselines fail.

Question: *Who is Harriet Hemings's maternal grandfather?* **Answer:** John Wayles

Model	Setup	Response
T5-XXL	FT	**Ted Hughes**
	PT	**John Hemings**
KNIT5-XXL	FT	**Betty Hemings**
	PT	**John Hemings**
	SPoT	Harriet Hemings ; mother ; Sally Hemings ; father ; **Thomas Hemings**
	PATH	**PP:** Harriet Hemings ; mother ; father
		HP: Harriet Hemings ; mother ; Sally Hemings ; father ; **John Wayles**
	MIXHOP	Harriet Hemings ; mother ; Sally Hemings ; father ; **John Wayles**

Table 9: An example case where all baselines fail, and additionally SPoT (Vu et al., 2022) also produces the incorrect final entity, but our two proposed methods succeed.

Question: *Who is Christopher Blom Paus's aunt?* **Answer:** Hedevig Christine Paus

Model	Setup	Response
T5-XXL	FT	**Clotilde of Saxe - Lauenburg**
	PT	**Annemarie Blom Paus**
KNIT5-XXL	FT	**Anna of Oldenburg**
	PT	**Christina Paus**
	SPoT	Christopher Blom Paus ; father ; Ole Paus ; sibling ; **Kjersti Bua Paus**
	PATH	**PP:** Christopher Blom Paus ; mother ; sibling
		HP: Christopher Blom Paus ; mother ; Margrete Laarmann ; sibling ; **Kjartan Flóki**
	MIXHOP	Christopher Blom Paus ; mother ; Ulla Blom ; sibling ; **Gunnar Blom**

Table 10: An example of an ambiguous question (since "aunt" can be father's sister or mother's sister) on which all approaches fail. Importantly, methods that use random-walks accurately generate the relations required to answer the question, but fail at predicting the correct entities.

cases, we randomly sample a template from the entire set, equally weighing each. In total, we end up with 83,643 train, 5,022 validation, and 6,440 test QA pairs.

Question: *What nationality is John Bede Dalley's father ?* **Answer:** United Kingdom		
Model	**Setup**	**Response**
T5-XXL	FT	**British**
	PT	**British**
KNIT5-XXL	FT	**English**
	PT	**English**
	SPOT	John Bede Dalley ; father ; William Dalley ; country of citizenship ; **English**
	PATH	**PP:** John Bede Dalley ; father ; country of citizenship **HP:** John Bede Dalley ; father ; William Bede Dalley ; country of citizenship ; **English**
	MIXHOP	John Bede Dalley ; father ; William Dalley, 1st Viscount Darnley ; country of citizenship ; **British**

Table 11: An example of a scenario where all models fail at answering the question correctly, but this is likely attributable to the dataset since it does not contain aliases.

Relation	**Template Space**	**Relation**	**Template Space**
director	*Who is the director of X?, Who directed the film X?*	mother	*Who is the mother of X?, Who is X's mother?*
date of birth	*What is the date of birth of X?, When is X's birthday?, When was X born?*	founded by	*Who is the founder of X?, Who founded X?*
date of death	*When did X die?, What is the date of death of X?*	inception	*When was X founded?*
country	*What country is X from?, What is the nationality of X?*	manufacturer	*Who manufactures X?*
country of citizenship	*What country is X from?, What is the nationality of X?*	performer	*Who is the performer of the song X?, Who performed the song X?*
award received	*What is the award that X received?, Which award did X receive?*	place of birth	*Where was X born?, What is the place of birth of X?*
cause of death	*Why did X die?, What was the cause of X's death?*	place of burial	*Where was X buried?, Where is the place of burial of X?*
composer	*Who is the composer of X?, Who composed X?*	place of death	*Where did X die?, Where is the place of death of X?*
creator	*Who is the creator of X?, Who created X?*	place of detention	*Where did X go to prison?, Where was X detained?*
child	*Who is the child of X?*	presenter	*Who is the presenter of X?, Who presented X?*
doctoral advisor	*Who is the doctoral advisor of X?*	publisher	*Who published X?, What company published X?*
editor	*Who is the editor of X?, Who edited X?*	sibling	*Who is the sibling of X?, Who is X's sibling?*
educated at	*Where did X graduate from?, What is the alma mater of X?, Where did X study?*	spouse	*Who is the spouse of X?, Who is X's spouse?*
employer	*Who is the employer of X?, Where does X work?*	student of	*Who was the teacher of X?, Who was X's teacher?*
father	*Who is the father of X?, Who is X's father?*		

Table 12: Question templates for for each of the 29 relations, used to create 1WikiHopQA. *X* stands for the subject.

ACL 2023 Responsible NLP Checklist

A For every submission:

☑ A1. Did you describe the limitations of your work?
Section 6

☑ A2. Did you discuss any potential risks of your work?
Section 6 (under limitations)

☑ A3. Do the abstract and introduction summarize the paper's main claims?
Intro: Section 1

☒ A4. Have you used AI writing assistants when working on this paper?
Left blank.

B ☑ Did you use or create scientific artifacts?

Section 3.1 (we repurposed an existing dataset) Section 2 and 4 (we created new prompting techniques that lead to new instances of existing models)

☑ B1. Did you cite the creators of artifacts you used?
Section 3

☑ B2. Did you discuss the license or terms for use and / or distribution of any artifacts?
Footnote 1

☒ B3. Did you discuss if your use of existing artifact(s) was consistent with their intended use, provided that it was specified? For the artifacts you create, do you specify intended use and whether that is compatible with the original access conditions (in particular, derivatives of data accessed for research purposes should not be used outside of research contexts)?
The original authors of the dataset we used did not provide any instructions for intended use. However the artifacts in this work were used for research purposes only.

☒ B4. Did you discuss the steps taken to check whether the data that was collected / used contains any information that names or uniquely identifies individual people or offensive content, and the steps taken to protect / anonymize it?
We skipped this since we repurposed an existing QA dataset (https://aclanthology.org/2020.coling-main.580/), released under the Apache 2.0 license, which contains questions and answers about entities and relations sourced from Wikidata, which does not contain any sensitive information about individual people.

☑ B5. Did you provide documentation of the artifacts, e.g., coverage of domains, languages, and linguistic phenomena, demographic groups represented, etc.?
See Section C (Appendix). We discussed number of relations and entities (i.e., coverage of domains) in our repurposed version of an existing dataset. We also explicitly mention that the dataset is in English, and also provide a table of the unique templates used for generating questions.

☑ B6. Did you report relevant statistics like the number of examples, details of train / test / dev splits, etc. for the data that you used / created? Even for commonly-used benchmark datasets, include the number of examples in train / validation / test splits, as these provide necessary context for a reader to understand experimental results. For example, small differences in accuracy on large test sets may be significant, while on small test sets they may not be.
Section 3.1 and Appendix C

The Responsible NLP Checklist used at ACL 2023 is adopted from NAACL 2022, with the addition of a question on AI writing assistance.

C ☑ Did you run computational experiments?

Section 3 and 4

☑ C1. Did you report the number of parameters in the models used, the total computational budget (e.g., GPU hours), and computing infrastructure used?
Appendix A

□ C2. Did you discuss the experimental setup, including hyperparameter search and best-found hyperparameter values?
No response.

☑ C3. Did you report descriptive statistics about your results (e.g., error bars around results, summary statistics from sets of experiments), and is it transparent whether you are reporting the max, mean, etc. or just a single run?
Appendix A

☑ C4. If you used existing packages (e.g., for preprocessing, for normalization, or for evaluation), did you report the implementation, model, and parameter settings used (e.g., NLTK, Spacy, ROUGE, etc.)?
Appendix A (metric calculation with default T5x implementation)

D ☒ Did you use human annotators (e.g., crowdworkers) or research with human participants?

Left blank.

□ D1. Did you report the full text of instructions given to participants, including e.g., screenshots, disclaimers of any risks to participants or annotators, etc.?
No response.

□ D2. Did you report information about how you recruited (e.g., crowdsourcing platform, students) and paid participants, and discuss if such payment is adequate given the participants' demographic (e.g., country of residence)?
No response.

□ D3. Did you discuss whether and how consent was obtained from people whose data you're using/curating? For example, if you collected data via crowdsourcing, did your instructions to crowdworkers explain how the data would be used?
No response.

□ D4. Was the data collection protocol approved (or determined exempt) by an ethics review board?
No response.

□ D5. Did you report the basic demographic and geographic characteristics of the annotator population that is the source of the data?
No response.