Faithful Knowledge Graph Explanations in Commonsense Question Answering

Guy Aglionby and **Simone Teufel**
Department of Computer Science and Technology
University of Cambridge
United Kingdom
{guy.aglionby,sht25}@cl.cam.ac.uk

Abstract

Knowledge graphs are commonly used as sources of information in commonsense question answering, and can also be used to express explanations for the model's answer choice. A common way of incorporating facts from the graph is to encode them separately from the question, and then combine the two representations to select an answer. In this paper, we argue that highly faithful graph-based explanations cannot be extracted from existing models of this type. Such explanations will not include reasoning done by the transformer encoding the question, so will be incomplete. We confirm this theory with a novel proxy measure for faithfulness and propose two architecture changes to address the problem. Our findings suggest a path forward for developing architectures for faithful graph-based explanations.

1 Introduction

In commonsense question answering, many approaches incorporate knowledge from external resources in addition to using large pre-trained language models. Most often this is done to improve performance, however explanations can also be expressed by highlighting a subset of this information. Making the model output the facts used to answer a particular question can increase trustworthiness and help with debugging.

For this kind of explanation to be helpful, it must faithfully represent the model's reasoning; that is, the explanation must accurately reflect the facts used to answer the question. Faithfulness is independent of whether or not the explanation is a reasonable justification of why an answer was chosen (Jacovi and Goldberg, 2020). We refer to how convincing an explanation is for an answer as plausibility.[1] It is useful to separate the two concepts as a faithful representation of the model's reasoning process will be the most useful to a developer who is debugging a model, regardless of whether it is plausible or not.

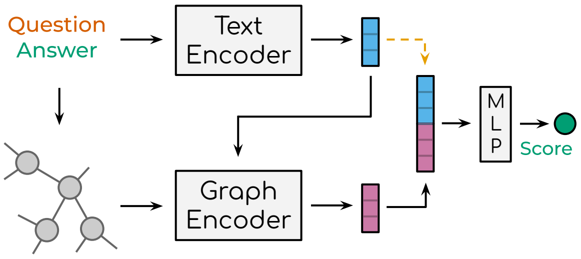

Figure 1: The architecture of one class of question answering models that use knowledge graphs. The orange dashed line is removed in our first ablation.

We argue that explanations from a broad class of models are of limited faithfulness (K M et al., 2018; Mihaylov and Frank, 2018; Lin et al., 2019; Feng et al., 2020; Yasunaga et al., 2021; Madaan et al., 2021, illustrated in figure 1). Explanations extracted from the graph encoder (figure 2) are unlikely to reflect the full set of facts used because the text encoder also independently reasons about the question and contributes to answer selection.

In addition to our theoretical arguments, we propose a proxy method for empirically measuring faithfulness. Our results confirm that, because performance does not significantly change when unhelpful input is given to the graph encoder, it has minimal influence on the final answer choice. Explanations extracted from the graph encoder therefore should not be used as they are not faithful to the reasoning of the overall model.

We propose two changes to the model architecture and find that they increase the proportion of reasoning done in the graph encoder, thereby also increasing explanation faithfulness. The results also reflect a difference we identify in how two graph encoders interact with the text encoder, leading to a path forward for developing architectures for faithful explanations.

[1] Terminology for these concepts varies, e.g. faithfulness is also known as trustworthiness. See Jacovi and Goldberg (2020) for discussion.

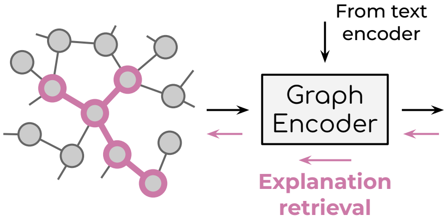

Figure 2: An explanation retrieval technique applied to the graph encoder selects nodes and edges from the input graph to be the explanation.

2 Model architecture

We focus on a class of models that combine a text encoder and a graph encoder for commonsense question answering (figure 1). Explanations are naturally expressed as a subset of edges in the input graph (figure 2) and could be used for understanding which facts the model used to choose its answer. These explanations are easy to interpret. Alternative approaches to incorporating external knowledge exist, like serialising knowledge graph triples and concatenating them to the text encoder input. A common approach to explaining predictions of text encoders is to calculate token-level attributions (Ribeiro et al., 2016), which is too specific in this scenario where facts are the desired output.

We test the behaviour of two recent models: MHGRN (Feng et al., 2020) and QA-GNN (Yasunaga et al., 2021). The high-level operation of both models is comparable and representative of others that have the same architecture. An embedding of the question context – a question with an answer choice – is obtained from the text encoder, which in both cases is `RoBERTa-Large` (Liu et al., 2019). The context is also used to extract a subgraph of up to 200 nodes (sometimes called a schema graph) from a larger knowledge graph.

The graph encoder creates an embedding for the knowledge in this subgraph; a message passing graph neural network (GNN) (Gilmer et al., 2017) is used in both cases. A key difference between the two models is the structure of the GNN. In each layer of MHGRN, multiple paths through the subgraph are found that end at each node, which are encoded and then pooled to form the new node embedding. QA-GNN instead updates nodes at each layer by aggregating messages from direct neighbours. After the final layer in both models, the node embeddings are combined via attentional pooling with respect to the text embedding.

The models also use the text embedding at each layer in different ways. In MHGRN, it is used to calculate the relevance of each path when creating the new embedding for each node. In QA-GNN, a pseudo-node initialised with this embedding is added to the graph, allowing it to participate in message passing with the other nodes.

To score each answer choice, the text embedding of the question context is concatenated with the embedding of that answer's extracted subgraph. An MLP is then used to produce a score.

3 Explanation faithfulness

We claim that the class of models described in §2 are intrinsically unable to provide graph-structured explanations that are highly faithful to the full model. Our desire for these explanations is that they are the collection of facts used by the model to complete a natural language understanding task. The more faithful these explanations are, the more useful they will be for developers to understand model behaviour. Following Jacovi and Goldberg (2020), we do not consider faithfulness to be a binary quality. Instead, we argue that different architectural choices either increase or decrease how faithful extracted explanations can be.

Explanations have low faithfulness because this class of model uses both text and graph embeddings to choose between answer candidates, but explanations are only extracted from the graph encoder. These explanations cannot give a full characterisation of the facts used to answer the question, as the text encoder is likely to have also contributed relevant information. When the text encoder is a pre-trained transformer it is infeasible that it does not contribute in this way, as they can achieve reasonable performance on question answering (Devlin et al., 2019). Graph-structured explanations are therefore necessarily incomplete.

3.1 Increasing faithfulness

We highlight three aspects of these models that, if changed, would increase explanation faithfulness.

Use of text embedding The text embedding is directly used for prediction in the final MLP of the model, represented as the orange dashed line in figure 1. This is the clearest way in which the graph encoder is skipped, and we argue that this must be removed if graph-based explanations are to be used for understanding models.

The text embedding is also used in the graph component. This inclusion is unavoidable, as it is necessary to provide the question context to guide reasoning in the graph. However, the embedding could also transfer the results of reasoning carried out in the text encoder, which would not be represented in a graph explanation.

Freezing the text encoder A further way to minimise the influence of the text encoder is to freeze it during training. The model will still produce a meaningful representation of the question context, but with minimal task-specific information. Indeed, we found that embeddings from our pre-trained text encoder (`RoBERTa-Large`) with no fine-tuning gave just above chance accuracy on our evaluation datasets. Using these embeddings instead of fine-tuned ones necessarily increases faithfulness as the graph component now contributes to a higher proportion of reasoning.

Freezing the text encoder's weights will increase the faithfulness of explanations from QA-GNN more than from MHGRN due to differences in how they use the text embedding. In MHGRN, information from the text encoder is only expressed as weights for explicit reasoning chains, so any reasoning can be reflected in a graph-structured explanation. It is not clear whether freezing the text encoder is required in this case. For QA-GNN however, the text embedding participates in message passing with nodes, so the text encoder must be frozen.

Interpretability technique Although we do not investigate them here, it is also important that the explanation extraction technique has been evaluated for faithfulness (figure 2). MHGRN and QA-GNN both use attention values to select a subset of edges to be in the explanation. However, it is not clear that attention faithfully represents how models reason (Jain and Wallace, 2019; Serrano and Smith, 2019). Approaches which specifically consider faithfulness, like GraphMask (Schlichtkrull et al., 2021), should be considered instead.

3.2 A proxy for faithfulness

In addition to our theoretical arguments, we propose a proxy technique to examine the impact of our proposed architecture changes. Human judgements are unsuitable for this purpose (Jacovi and Goldberg, 2020), but it is useful to quantify faithfulness particularly because we view it as a scale rather than binary property.

Our method measures how much of the reasoning in the overall model is performed in the graph encoder. If assumption 1 does not hold, it suggests that reasoning is being performed in a different part of the model, which would not be captured in the explanation. The explanation would therefore not be faithful to the full model's workings.

Assumption 1. If a faithful explanation is to be extracted from a graph encoder, large changes to its input should have a large impact on model behaviour.

The large change we make to the graph encoder input is to shuffle schema graphs across questions so they become irrelevant. If a large proportion of reasoning is done in the graph encoder, this shuffling should cause accuracy to be at or below random chance. This is because the shuffled graph contains at best minimal useful information, and at worst misleading information for answering the question correctly. If reasoning is instead predominantly done by the text encoder, we would expect accuracy to drop less severely. Therefore the closer to random chance the accuracy is, the more faithful the explanations are.

We combine the regular and shuffled data conditions with three model conditions: the unmodified versions of MHGRN and QA-GNN, and versions with the two successive ablations from §3.1. The first ablation removes the textual embedding from the final MLP only (− *Embed.*). On top of this, the second also freezes the weights of the text encoder (− *Train TE.*).

3.3 Training and data

We use the same training hyperparameters as Feng et al. (2020) and Yasunaga et al. (2021), which we reproduce in appendix A. We slightly modify the official code for each model to implement our architecture changes.[2] We use ConceptNet (Speer et al., 2017) as the base knowledge graph, and evaluate on CommonsenseQA (CSQA) (Talmor et al., 2019) and OpenBookQA (OBQA) (Mihaylov et al., 2018). We use standard dataset splits for OBQA, and 'in house' (Lin et al., 2019) splits for CSQA. We use `RoBERTa-Large` as our text encoder, which we also use to initialise node embeddings in the GNN following Feng et al. (2020). We re-train the model for each experiment with 10 different random seeds and report the mean accuracy.

[2] We release our code at `https://github.com/GuyAglionby/faithful-kg-qa-explanations`.

4 Results

Table 1 gives our experimental results; supplementary details are provided in appendix B. Following Reimers and Gurevych (2017), we use the Kolmogorov-Smirnov test (Massey, 1951) to check whether the test score distributions for each pair of model-data setups are significantly different. In this section, we compare the results in the two data scenarios in each of the model ablation scenarios. For both datasets, we expect that if a randomly chosen graph is used then a system that can faithfully output explanations will have accuracy at or below chance (25%).

	CSQA		OBQA	
	Reg.	*Shuf.*	*Reg.*	*Shuf.*
QA-GNN	70.26	69.72	62.98	65.24
− *Embed.*	64.68	60.68	52.36	53.66
− *Train TE.*	30.46	19.50	40.70	25.26
MHGRN	69.71	69.07	65.98	65.50
− *Embed.*	24.66	19.64	42.56	31.96
− *Train TE.*	24.45	19.76	41.04	36.00

Table 1: Average accuracy (10 random seeds) in two data scenarios *regular* and *shuffled* for each dataset, and three model scenarios for each model type.

Original model In all four cases, the accuracy in the shuffled scenario remains high: there is no significant difference with the regular case. Assumption 1 is contradicted, and this result suggests that the model architecture allows the text encoder to do all of the reasoning. Even in the regular scenario it is likely that a large proportion of the reasoning is done outside of the graph encoder because the change in performance between the two data scenarios is so low. We conclude that explanations extracted from the graph encoder do not reflect the overall model's operation.

Text embedding removed We next examine the models where the text embedding is no longer included in the final MLP. Here there is a significant ($p < 0.02$) difference between accuracy when using regular versus shuffled graphs in all cases but QA-GNN on OBQA. The fact that unhelpful input now causes performance to drop suggests that more reasoning is done in the graph component in this model setup than in the previous one. Explanations from the graph encoder are therefore more faithful to overall model behaviour.

Although faithfulness has increased, the shuffled accuracy for QA-GNN is still substantially above random. This is likely due to how the text embedding is used within its graph encoder. Explanations from the graph encoder are therefore still largely unfaithful to the model.

The MHGRN shuffled result is much nearer random chance. Here the text embedding is only used to calculate attention weights, so it is more difficult for reasoning in the text encoder to make up for the unhelpful input to the graph encoder. This change, therefore, increases faithfulness in MHGRN substantially more than in QA-GNN.

It is surprising that MHGRN's regular CSQA result is also near chance. This suggests either that MHGRN is unable to learn how to use the graph with this change made, or that the schema graph does not contain useful information for the task. We conclude that the second case is more likely because the model is still able to achieve 42.56% accuracy on OBQA.

Both ablations When both ablations are applied, three of the four models have a significant difference in accuracy between the regular and shuffled scenarios ($p < 0.001$), and the shuffled performance is at or below random chance. The difference is not significant for MHGRN on OBQA, although an absolute drop does occur. We further note the significant ($p < 0.01$) change in accuracy for all four QA-GNN models from − *Embed* to − *Train TE.*, which confirms that much of the previous performance is a result of the text encoder learning the task. These results suggest that the graph encoder is now the model component with the highest influence on performance, which is the ideal case for retrieving faithful explanations.

None of the differences in accuracy between − *Embed* and − *Train TE.* for MHGRN are significant. The lack of change suggests that the text encoder already had minimal influence on accuracy in the first ablation, so there is no need to further reduce it with the second.

There is one anomalous change in MHGRN's performance, where accuracy increases from 31.96% to 36.00% on OBQA. As there is unlikely to be a meaningful way to combine nodes from a random schema graph for a question, the training signal for the graph encoder as it performs attentive pooling is likely to be noisy. The same is true for the pooling of paths for new node embeddings. When the text encoder is frozen the influence of this noise is removed, which may explain the rise in performance.

5 Discussion

We have demonstrated that in MHGRN and QA-GNN, as the ability of the pre-trained text encoder to learn a task is curtailed, accuracy significantly decreases. This result suggests that it was the text encoder that most contributed to performance. Our finding is reinforced by the fact that shuffling subgraphs across questions has no significant impact on performance. Explanations extracted from the graph encoder will therefore be unfaithful to the overall operation of the model, as they will not capture the substantial reasoning done in the text encoder.

Our two successive model changes make explanations more faithful by increasing the proportion of reasoning done in the graph encoder. Removing the direct use of the text embedding in the final classification MLP is crucial for retrieving faithful explanations. For QA-GNN it is also necessary to freeze the text encoder weights; this is not the case for MHGRN due to how it uses the text embedding.

Our results generalise from the two models we examined. KagNet (Lin et al., 2019) and the Knowledgeable Reader (Mihaylov and Frank, 2018) precede both models and work most similarly to MHGRN. There, the question context embedding is also used to calculate attention weights for embeddings of different facts. GreaseLM (Zhang et al., 2022) instead builds on QA-GNN, and further integrates the text encoder embeddings with the GNN. Although the authors do not discuss explanations with this model, it is an example of where further integration of the two encoders would harm faithfulness.

Future work on faithfully interpretable models may structure their use of the text encoder in a similar way to MHGRN, as this more easily leads to faithful explanations. Such work might examine how to better train the text encoder to weight the relevancy of facts. Although recent work successfully achieves higher accuracy via tighter coupling of the graph and text encoders (Zhang et al., 2022), models that maintain separation are likely to yield faithful explanations more easily. How to do this while maintaining reasonable accuracy remains an open question, although we argue that the ability to produce faithful explanations is particularly valuable. One cause for the performance drop might be the quality of the extracted subgraphs: our results suggest that they are more appropriate for CommonsenseQA than OpenBookQA.

Faithful explanations can be used to better understand model behaviour to help improve them. An implausible but faithful explanation is a signal to a model developer that something may need to be changed in the model. In this way, explanation quality can also be used to judge how good a model is, alongside accuracy. However, the plausibility of an explanation is only a useful property if the explanation is faithful.

Limitations

Although we have argued for why using random schema graphs is a reasonable method for investigating the faithfulness of explanations retrieved from this class of model, they remain a proxy. It is possible therefore that for some of our model ablation conditions we draw conclusions about faithfulness that are too strong or too weak.

Additionally, we only investigate two models from a class of architectures for incorporating external knowledge and do not examine other kinds of methods, as discussed in §2. Although we argue that our analysis is sound for this class, it is possible that another general type of architecture would be more suited to obtaining faithful explanations for question answering. Our model changes cause accuracy to drop substantially, although we argue that the ability to produce faithful explanations is valuable.

Acknowledgements

We would like to thank Christopher Davis for feedback on an earlier draft of this paper. We also thank the anonymous reviewers for their questions and comments.

References

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.

Yanlin Feng, Xinyue Chen, Bill Yuchen Lin, Peifeng Wang, Jun Yan, and Xiang Ren. 2020. Scalable Multi-Hop Relational Reasoning for Knowledge-Aware Question Answering. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1295–1309, Online. Association for Computational Linguistics.

Justin Gilmer, Samuel S. Schoenholz, Patrick F. Riley, Oriol Vinyals, and George E. Dahl. 2017. Neural Message Passing for Quantum Chemistry. In *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 1263–1272. PMLR.

Alon Jacovi and Yoav Goldberg. 2020. Towards Faithfully Interpretable NLP Systems: How Should We Define and Evaluate Faithfulness? In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4198–4205, Online. Association for Computational Linguistics.

Sarthak Jain and Byron C. Wallace. 2019. Attention is not Explanation. 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 3543–3556, Minneapolis, Minnesota. Association for Computational Linguistics.

Annervaz K M, Somnath Basu Roy Chowdhury, and Ambedkar Dukkipati. 2018. Learning beyond Datasets: Knowledge Graph Augmented Neural Networks for Natural Language Processing. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 313–322, New Orleans, Louisiana. Association for Computational Linguistics.

Bill Yuchen Lin, Xinyue Chen, Jamin Chen, and Xiang Ren. 2019. KagNet: Knowledge-Aware Graph Networks for Commonsense Reasoning. 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 2829–2839, Hong Kong, China. Association for Computational Linguistics.

Liyuan Liu, Haoming Jiang, Pengcheng He, Weizhu Chen, Xiaodong Liu, Jianfeng Gao, and Jiawei Han. 2020. On the Variance of the Adaptive Learning Rate and Beyond. In *Eighth International Conference on Learning Representations*, Online.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. RoBERTa: A Robustly Optimized BERT Pretraining Approach. *arXiv:1907.11692*.

Aman Madaan, Niket Tandon, Dheeraj Rajagopal, Peter Clark, Yiming Yang, and Eduard Hovy. 2021. Think about it! Improving defeasible reasoning by first modeling the question scenario. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 6291–6310, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Frank J. Massey. 1951. The Kolmogorov-Smirnov Test for Goodness of Fit. *Journal of the American Statistical Association*, 46(253):68–78.

Todor Mihaylov, Peter Clark, Tushar Khot, and Ashish Sabharwal. 2018. Can a Suit of Armor Conduct Electricity? A New Dataset for Open Book Question Answering. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2381–2391. Association for Computational Linguistics.

Todor Mihaylov and Anette Frank. 2018. Knowledgeable Reader: Enhancing Cloze-Style Reading Comprehension with External Commonsense Knowledge. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 821–832, Melbourne, Australia. Association for Computational Linguistics.

Nils Reimers and Iryna Gurevych. 2017. Reporting Score Distributions Makes a Difference: Performance Study of LSTM-networks for Sequence Tagging. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 338–348, Copenhagen, Denmark. Association for Computational Linguistics.

Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. 2016. "Why Should I Trust You?": Explaining the Predictions of Any Classifier. In *Proceedings of the 22Nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '16, pages 1135–1144, New York, NY, USA. ACM.

Michael Sejr Schlichtkrull, Nicola De Cao, and Ivan Titov. 2021. Interpreting Graph Neural Networks for NLP With Differentiable Edge Masking. In *Ninth International Conference on Learning Representations*, Online.

Sofia Serrano and Noah A. Smith. 2019. Is Attention Interpretable? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2931–2951, Florence, Italy. Association for Computational Linguistics.

Robyn Speer, Joshua Chin, and Catherine Havasi. 2017. ConceptNet 5.5: An Open Multilingual Graph of General Knowledge. In *Thirty-First AAAI Conference on Artificial Intelligence*.

Alon Talmor, Jonathan Herzig, Nicholas Lourie, and Jonathan Berant. 2019. CommonsenseQA: A Question Answering Challenge Targeting Commonsense Knowledge. 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 4149–4158, Minneapolis, MN. Association for Computational Linguistics.

Michihiro Yasunaga, Hongyu Ren, Antoine Bosselut, Percy Liang, and Jure Leskovec. 2021. QA-GNN: Reasoning with Language Models and Knowledge Graphs for Question Answering. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 535–546, Online. Association for Computational Linguistics.

Xikun Zhang, Antoine Bosselut, Michihiro Yasunaga, Hongyu Ren, Percy Liang, Christopher D Manning, and Jure Leskovec. 2022. GreaseLM: Graph REASoning Enhanced Language Models. In *Tenth International Conference on Learning Representations*, Online.

A Hyperparameters

We train both models for a maximum of 70 epochs, and because we observed high variability across random seeds we use an early stopping patience of 30.

QA-GNN All parameters optimised with RAdam (Liu et al., 2020). Batch size is 128. A maximum of 128 tokens are input to the text encoder, which is trained with learning rate $1\mathrm{e}-5$ but frozen for the first 4 epochs. The 5-layer GNN has 200-dimensional embeddings and is trained with learning rate $1\mathrm{e}-3$. Parameters have L2 weight decay of 0.01 applied.

MHGRN All parameters optimised with RAdam. Batch size is 32. A maximum of 128 tokens are input to the text encoder, which is trained with learning rate $1\mathrm{e}-5$ but frozen for the first 3 epochs. The 1-layer GNN has 100-dimensional embeddings and is trained with learning rate $1\mathrm{e}-3$. Each layer performs 3-hop message passing. Parameters have L2 weight decay of 0.01 applied.

B Additional results

Standard deviations on the test set for each experiment are given in table 2, and development set scores in table 3. The average run times of these experiments on an Nvidia A100 GPU are shown in table 4, which correspond to the number of optimisation steps in table 5.

	CSQA		OBQA	
	Reg.	*Shuf.*	*Reg.*	*Shuf.*
QA-GNN	1.02	0.53	3.03	2.90
− *Embed.*	1.42	2.26	10.12	6.75
− *Train TE.*	1.19	1.32	2.36	2.28
MHGRN	0.73	0.91	2.48	1.68
− *Embed.*	0.79	1.12	6.60	13.18
− *Train TE.*	0.68	1.31	2.37	10.11

Table 2: Standard deviation of test set score across 10 runs, corresponding to table 1.

	CSQA		OBQA	
	Reg.	*Shuf.*	*Reg.*	*Shuf.*
QA-GNN	76.11	75.82	65.36	68.08
− *Embed.*	74.13	70.76	56.78	59.02
− *Train TE.*	33.49	22.27	45.62	28.30
MHGRN	75.24	75.28	69.52	69.24
− *Embed.*	29.37	22.44	47.76	36.68
− *Train TE.*	29.07	22.29	44.82	42.08

Table 3: Average development set accuracy across 10 runs, used to select the test scores reported in table 1.

	CSQA		OBQA	
	Reg.	*Shuf.*	*Reg.*	*Shuf.*
QA-GNN	3.29	2.92	2.12	2.07
− *Embed.*	4.21	4.06	1.96	2.07
− *Train TE.*	1.41	1.56	0.97	1.01
MHGRN	2.23	2.23	1.85	1.72
− *Embed.*	2.02	2.90	1.25	1.50
− *Train TE.*	2.04	2.72	1.27	1.45

Table 4: Average run time (in hours) for experiments in table 1.

	CSQA		OBQA	
	Reg.	*Shuf.*	*Reg.*	*Shuf.*
QA-GNN	3524	3162	2769	2683
− *Embed.*	4509	4348	2582	2718
− *Train TE.*	2539	2827	2204	2293
MHGRN	10,906	10,826	10,881	10,168
− *Embed.*	9922	13,992	7487	8975
− *Train TE.*	10,028	13,273	7564	8634

Table 5: Average number of optimisation steps for experiments in table 1.