KGQA Without Retraining

Nick McKenna
University of Edinburgh
nick.mckenna@ed.ac.uk

Priyanka Sen
Amazon Alexa AI
sepriyan@amazon.co.uk

Abstract

Popular models for Knowledge Graph Question Answering (KGQA), including semantic parsing and End-to-End (E2E) models, decode into a constrained space of KG relations. Although E2E models accommodate novel entities at test-time, this constraint means they cannot access novel relations, requiring expensive and time-consuming retraining whenever a new relation is added to the KG. We propose KG-Flex, a new architecture for E2E KGQA that instead decodes into a continuous embedding space of relations, which enables use of novel relations at test-time. KG-Flex is the first to support KG updates with entirely novel triples, free of retraining, while still supporting end-to-end training with simple, weak supervision of (Q, A) pairs. Our architecture saves on time, energy, and data resources for retraining, yet we retain performance on standard benchmarks. We further demonstrate zero-shot use of novel relations, achieving up to 82% of baseline hit@1 on three QA datasets. KG-Flex can also fine-tune, requiring significantly shorter time than full retraining; fine-tuning on target data for 10% of full training increases hit@1 to 89-100% of baseline.

1 Introduction

Knowledge Graph Question Answering (KGQA) is the task of answering questions using facts in a Knowledge Graph (KG). Common approaches to KGQA include semantic parsing (Rongali et al., 2020) and End-to-End (E2E) Question Answering techniques (Cohen et al., 2020). E2E approaches are promising due to being composed of entirely differentiable operations, including program prediction and execution using a Differentiable KG (DKG), and the ease of training with simple (question, answer) pairs. However, these methods decode into a constrained space of KG relations which are then used to traverse the KG. While this works well for benchmark datasets where the KGs are static, it fails to scale to real use cases where KGs are frequently updated.

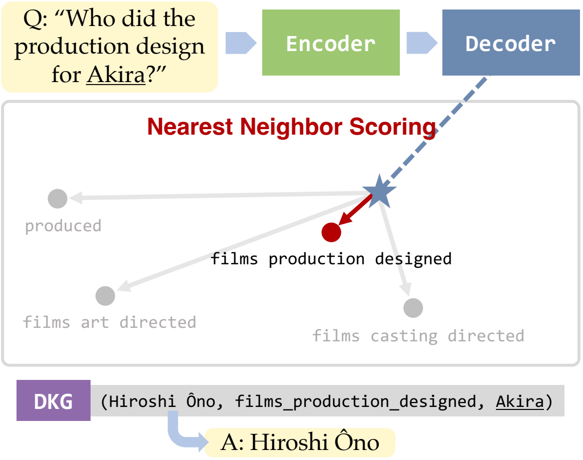
Q: "Who did the production design for Akira?"
Encoder
Decoder
Nearest Neighbor Scoring
produced
films production designed
films art directed
films casting directed
DKG
(Hiroshi Ôno, films_production_designed, Akira)
A: Hiroshi Ôno

Figure 1: KG-Flex enables KG updates with new entities or relations at test-time without retraining. Given a question, an embedding is predicted in the space of (pre-computed) KG relation embeddings. Available relations are scored relatively by distance to prediction.

For example, Wikidata (Vrandečić and Krötzsch, 2014) is a commonly used Knowledge Graph that is actively updated. Between March 2022 and March 2023, the number of reported properties (relations in Wikidata) increased from 9.7K to 10.9K. At the time of writing, there are almost 200 new properties proposed for addition, including relations about new platforms or services (e.g., *Patreon user ID*, *Peacock ID*), and relations improving existing ontologies (e.g., *Pokemon category*, *alternate universe counterpart*).

In order to handle new relations, most KGQA methods require full retraining to learn a new output space of possible relations. These methods also require additional training data with examples using the new relations. We argue that incrementally updating the KG should not require full model retraining, a mostly redundant process which is energy- and time-intensive.

We present KG-Flex, an E2E model architecture that overcomes this problem by instead decoding

Proceedings of The Fourth Workshop on Simple and Efficient Natural Language Processing (SustaiNLP), pages 212–218
July 13, 2023 ©2023 Association for Computational Linguistics

into an open embedding space in which relations are expressed in natural language. Given an input question, the model predicts the answer relation *embedding*, and available triples in the KG are scored against the prediction via their relations (example in Figure 1). KG-Flex is the first End-to-End KGQA model that allows updates to both KG entities and relations at test-time, without retraining.

We show that KG-Flex retains performance on standard benchmarks compared to a similar model, while demonstrating additional capabilities. In a zero-shot setting in three QA datasets, KG-Flex scores between 40-82% of baseline hit@1 on questions using relations that were held out during training time, a task which is impossible for previous models using a fixed decoder. Further, by fine-tuning for 10% of full training, scores are increased to 89-100% of baseline.

2 Related Work

Traditional approaches to KGQA involve semantic parsing of natural language into logical forms. Semantic parsing models use a constrained decoder over the output space of symbols. Since models such as Rongali et al. (2020) treat relations as whole symbols, adding new relations requires increasing the decoder output size and retraining the model. Further, collecting new training data of natural language to logical forms (e.g., SPARQL) is expensive (Finegan-Dollak et al., 2018).

Other techniques transform queries and KG triples to an embedding space (Saxena et al., 2020; Sun et al., 2020). These methods do not require annotated KG queries, however, adding new relations requires retraining to update the model.

Recent End-to-End methods for KGQA (Cohen et al., 2020; Sen et al., 2021) are weakly supervised with (question, answer) pairs which, conditioned on a question, predict a probability distribution over KG relations. Execution on the KG involves following relations and returning probabilistically weighted answer entities. While E2E methods also do not require supervision of KG pathways, they still constrain relation decoding, which means that adding new relations requires retraining.

Previous work in expanding the flexibility of KGQA models at test-time include Ravishankar et al. (2021), using a two-step process of first predicting an intermediate query template, then adding KG-specific relations. However, this method still relies on expensive SPARQL query annotation for training data. Oguz et al. (2022) propose a method to unify structured and unstructured data sources by converting them all into text, however this sacrifices the useful structure of Knowledge Graphs.

3 KG-Flex

We introduce KG-Flex, a novel KGQA architecture that is designed for unconstrained relation decoding, and can be trained end-to-end using only weak supervision of questions and answers. KG-Flex is an encoder-decoder model using a Differentiable Knowledge Graph, in the family of End-to-End KGQA models such as ReifKB (Cohen et al., 2020) and Rigel-based models (Sen et al., 2021; Saffari et al., 2021). We make key changes in the decoder to enable the use of new relations at test-time. KG-Flex has 4 key stages.

3.1 Precompute KG Relation Embeddings

Ahead of train- or test-time, we pre-compute vector embeddings for all relations available in KG triples (in training and test, these are frozen). To do so, every KG relation $r_i \in \mathcal{R}$ is lightly preprocessed into natural language and encoded as a vector $\mathbf{r}_i \in \mathbb{R}^h$, $h = 768$, using the RoBERTa-base v2 Sentence Transformer (Reimers and Gurevych, 2019).

- **Freebase** (Bollacker et al., 2008) property IDs are preprocessed into the template "(type) property", e.g., *film.film_festival.location* ⇒ "(film festival) location"
- **Wikidata** property label text is already in natural language, e.g., "place of birth"
- **MetaQA** relations have underscores replaced with spaces, e.g., *directed_by* ⇒ "directed by"

3.2 Encode Question Text

At train- and test-time, a natural language question is encoded with RoBERTa-base v2 Sentence Transformer, into a vector representation $\mathbf{q} \in \mathbb{R}^h$. This is similar to earlier E2E models like Rigel (Sen et al., 2021), which uses a RoBERTa encoder.

3.3 Decode Relation Embedding

The decoder is the key improvement in KG-Flex, which predicts from amongst the relations available in the KG, unconstrained from a fixed schema. Conditioned on the question encoding $\mathbf{q}$, the KG-Flex decoder predicts an *embedding* for a relation which answers the question. All available relations are scored based on their Euclidean distance to

	WebQ	SimpleQ	MetaQA 1-hop	MetaQA 2-hop	MetaQA 3-hop
MemNN (Bordes et al., 2015)	22.7	61.6	–	–	–
KVMemNet (Miller et al., 2016)	46.7	–	95.8	25.1	10.1
GraftNet (Sun et al., 2018)	66.4	–	97.0	94.8	77.7
PullNet (Sun et al., 2019)	68.1	–	97.0	99.9	91.4
KBQA Adapter (Wu et al., 2019)	–	72.0	–	–	–
EmbedKGQA (Saxena et al., 2020)	66.6	–	97.5	98.8	94.8
TransferNet (Shi et al., 2021)	**71.4**	–	97.5	**100.0**	**100.0**
ReifKB (Cohen et al., 2020)	52.7	–	96.2	81.1	72.3
Rigel (Sen et al., 2021)	69.2	**79.9**	97.5	87.1	89.6
KG-Flex (ours)	68.9	79.2	**97.6**	90.1	87.2

Table 1: KG-Flex compared to baselines on three standard QA tasks: WebQuestions (WebQ), SimpleQuestions (SimpleQ), and MetaQA. Compared to the similar model Rigel, KG-Flex scores within 3 percentage points.

the prediction, and these scores are converted to a probability distribution via a softmax.

As in Sen et al. (2021) and Saffari et al. (2021), the decoding step is performed for T "hops" in the KG, where the hyperparameter T is fixed before training[1]. An attention mechanism is jointly learned as part of the model which predicts how many hops (up to T) is required for a given question, conditioned on $\mathbf{q}$. This is used to weight final entity predictions. For example, answering "What's the mascot of Obama's alma mater?" requires two hops from the entity *Obama*: first the ALMA-MATER relation, then MASCOT, so entities fetched in hop 2 will be weighted most heavily.

For each hop $t \in [1, T]$, we apply a decoder transformation $D_t \in \mathbb{R}^{(th)\times h}$ (with bias $\mathbf{b}_t \in \mathbb{R}^h$). D_t predicts a relation embedding $\mathbf{z}_t \in \mathbb{R}^h$, given the encoded question vector $\mathbf{q} \in \mathbb{R}^h$ and any predictions of earlier hops $\mathbf{z}_{<t}$.

$$\begin{aligned}\mathbf{z}_1 &= \mathit{tanh}(\mathbf{q}D_1 + \mathbf{b}_1)\\ \mathbf{z}_2 &= \mathit{tanh}([\mathbf{q};\mathbf{z}_1]D_2 + \mathbf{b}_2)\\ \mathbf{z}_3 &= \mathit{tanh}([\mathbf{q};\mathbf{z}_1;\mathbf{z}_2]D_3 + \mathbf{b}_3)\end{aligned}$$

The decoder is trained to predict a relation embedding which minimizes the Euclidean distance (L^2 norm) to the relation path that leads to the answer entity. Each decoder hop produces a probability distribution $\mathbf{d}_t$ over relations $r_i \in \mathcal{R}$ by a softmax over *negated* distances from $\mathbf{z}_t$ to precomputed $\mathbf{r}_i$:

$$\mathbf{d}_{t,i} = \frac{e^{-||\mathbf{z}_t - \mathbf{r}_i||_2}}{\sum_{j=0}^{|\mathcal{R}|} e^{-||\mathbf{z}_t - \mathbf{r}_j||_2}}$$

3.4 Execute on DKG

As an E2E model, KG-Flex executes a probabilistic query over its Differentiable KG (DKG) to produce weighted answer entities; these are scored to feed back the training signal through the model.

A DKG is just a re-representation of a KG as three matrices. In hop t, given a distribution over subjects $\mathbf{e}_t$ and relations $\mathbf{d}_t$, the "follow" operation (Cohen et al., 2020) computes a probability distribution over KG triple objects $\mathbf{e}_{t+1}$ using simple matrix multiplication:

$$\mathbf{e}_{t+1} = \mathit{follow}(\mathbf{e}_t, \mathbf{d}_t)$$

In the first hop, $\mathbf{e}_1$ is a one-hot vector of KG entities (question entity set to 1) [2]. For each hop $1 \leq t \leq T$, a probability distribution is predicted over KG entities, which are fed into the subsequent hop. The final model prediction is a distribution over KG entities discovered in all hops, weighted by the hop attention mechanism. During training, entity predictions are compared to the gold label entities via binary cross-entropy loss, and updates are backpropagated through the decoder and encoder.

4 Experiments

In our experiments, we use three datasets: **SimpleQuestions** (Bordes et al., 2015), a large-scale dataset of simple, one-hop questions based on FreeBase; **WebQuestionsSP** (Yih et al., 2016), a dataset of natural language questions containing up to 2 hops linked to FreeBase; and **MetaQA** (Zhang et al., 2018), a movies QA dataset divided into one, two, and three-hop subsets. MetaQA uses a KG that is internal to the dataset.

KG-Flex models are trained until dev set convergence or max 40,000 steps on a single NVIDIA Tesla V100 GPU (see Appendix A for details).

[1] We assume that $T = 3$ hops is sufficient to cover all realistic human questions.

[2] Like Cohen et al. (2020) and Sen et al. (2021), we begin with question entities pre-identified in the datasets.

		Full Test Set				Heldout Test Set			
		Rigel	**KG-Flex**			**Rigel**	**KG-Flex**		
Dataset	Domain	BL	BL	Zero-Shot	Fine-tuned	BL	BL	Zero-Shot	Fine-tuned
WebQuestions	film	69.2	68.9	65.8	68.7	71.4	76.8	31.3	76.8
WebQuestions	sports	69.2	68.9	66.0	69.0	49.7	53.6	28.4	51.0
SimpleQuestions	film	79.9	79.2	71.9	79.2	74.1	73.3	60.1	70.2
SimpleQuestions	medicine	79.9	79.2	77.5	79.7	79.4	82.2	60.2	72.8
MetaQA 1-hop	directed_by	97.5	97.6	92.6	97.6	97.1	97.2	70.5	97.1
MetaQA 2-hop	directed_by	87.1	90.1	79.7	90.6	90.1	85.7	54.8	86.7
MetaQA 3-hop	directed_by	89.6	87.2	62.9	85.7	89.2	86.2	36.9	85.5
MetaQA 1-hop	written_by	97.5	97.6	93.8	97.6	98.7	98.9	81.4	98.9
MetaQA 2-hop	written_by	87.1	90.1	81.7	91.3	87.1	90.0	68.0	91.7
MetaQA 3-hop	written_by	89.6	87.2	65.1	85.8	86.8	84.3	32.8	82.5

Table 2: For each dataset and domain, we evaluate three models: the **Baseline (BL)** is trained on the full dataset, **Zero-Shot** is trained with a **Domain** held out, and **Fine-tuned** is the Zero-Shot model fine-tuned for 4K steps on the full dataset. Each of these three models are evaluated on two datasets: **Full Test Set** (all examples in the test set), and **Heldout Test Set** (the subset of the test set using the held out relations).

4.1 Standard Benchmarks

First, we benchmark KG-Flex against existing methods on standard datasets. We report hit@1 scores, a metric that measures the percentage of questions where the highest probability entity predicted is correct[3]. Results are shown in Table 1. Compared to similar E2E models like Rigel, KG-Flex attains competitive performance within 3 percentage points.

4.2 Zero-Shot Transfer to Held-out Relations

We simulate a real-world scenario where new KG domains are added after training. We demonstrate how KG-Flex can predict using these held-out relations, an impossible task for prior E2E models like Rigel and ReifKB.

In training, we remove a subdomain of relations from the KG and all questions involving those relations from train and dev sets. Then at test-time we reintroduce the relations to the KG and include the held-out questions. We report on the full test set as well as the subset consisting of just the held-out questions. For SimpleQuestions, we remove all relations in the domains Film (61 relations) or Medicine (66 relations). For WebQuestions, we remove Film (24 relations) or Sports (45 relations). These domains represent a reasonably-sized KG update (< 10% of total relations). Since MetaQA contains only movie questions, we remove the "directed_by" or "written_by" relations (further info in Appendix B).

	Train	**Fine-tune**
WebQuestions	7 hr	45 min
SimpleQuestions	7 hr	1 hr
MetaQA 1-hop	1 hr	10 min
MetaQA 2-hop	1 hr	15 min
MetaQA 3-hop	1 hr	15 min

Table 3: Comparison of training time (for 40,000 steps) vs. fine-tuning (for 4,000 steps) on each of our datasets

Results are shown under **Zero-Shot** columns in Table 2. We compare to the Baseline KG-Flex results, which refer to training on the full dataset. On the held-out datasets the zero-shot models score up to 82% of the baseline score on simpler datasets (81.4 achieved / 98.9 baseline on written_by in MetaQA 1-hop; 60.1 / 73.3 on film in SimpleQuestions). However, they perform worse on more complex questions, reaching as low as 40% of the baseline on multi-hop question datasets (36.9 / 86.2 on directed_by in MetaQA 3-hop; 31.3 / 76.8 on film in WebQuestions). We attribute this to the compounding likelihood of error when predicting multiple relations at once. Since comparable models would score 0% on this task, we still consider this to be a valuable step forward.

4.3 Fine-tuned Transfer to Held-out Relations

To improve transfer to new relations, we further fine-tune the zero-shot models. Each zero-shot model is fine-tuned for 4,000 steps (10% of full training) on the entire dataset, including held-out relations. Results are under **Fine-tuned** in Table 2.

Our fine-tuned models are able to recapture between 89-100% of baseline KG-Flex performance on both the full test set and heldout test set. We

[3] Since SimpleQuestions may have multiple correct answers, we count predictions as correct if any entity within a tie for most probable is correct.

are able to fine-tune our model because we have an unconstrained decoder space, whereas comparable models such as ReifKB (Cohen et al., 2020) would require retraining from scratch. By fine-tuning for only a fraction of the full training time (see Table 3), we demonstrate that KG-Flex can efficiently adapt to new relations.

5 Conclusions

We present KG-Flex, a new model architecture for KGQA which is the first to use an unconstrained decoder over KG relations and to train end-to-end using simple (question, answer) pairs. While maintaining performance on benchmarks, KG-Flex is demonstrated to make use of incrementally changing live KGs without requiring expensive retraining. We show that KG-Flex uses novel relations added at test-time, handling simple questions with new relations in zero-shot, and handling more complex multihop questions by fine-tuning for only 10% of the training steps required for full retraining.

6 Limitations

We present KG-Flex, an end-to-end model that can access new relations at test-time without retraining. Our current KG-Flex model does not perform entity resolution, and so we rely on resolved entities provided by the datasets. However, resolved entities may not always be available, so tools such as automatic entity recognition may be necessary. While it is possible for end-to-end models to jointly learn to resolve entities in questions before relation following (Saffari et al., 2021), we consider this outside the scope of this focused work.

Additionally, KG-Flex is limited in the kinds of reasoning it can do over a Knowledge Graph. Currently, KG-Flex only performs relation following, so it cannot handle questions which require complex reasoning like counts, comparatives, min/max, etc, such as “Who is the tallest NBA player?” We hope to address this in future work.

Further, we test KG-Flex on popular datasets representing possible real human questions. However, we do not deeply investigate the semantic properties of these questions. Notably, McKenna and Steedman (2022) show that searching for similar relations in embedding space (as done in KG-Flex) may work better for paraphrastic inference, and only in certain cases for directional inference where semantic precision matters, e.g. DEFEAT entails PLAY, but PLAY does not entail DEFEAT. We leave deeper investigation of KG-Flex semantics and edge cases to future work.

References

Kurt Bollacker, Colin Evans, Praveen Paritosh, Tim Sturge, and Jamie Taylor. 2008. Freebase: A collaboratively created graph database for structuring human knowledge. In *Proceedings of the 2008 ACM SIGMOD International Conference on Management of Data*, SIGMOD ’08, page 1247–1250, New York, NY, USA. Association for Computing Machinery.

Antoine Bordes, Nicolas Usunier, Sumit Chopra, and Jason Weston. 2015. Large-scale simple question answering with memory networks.

William W. Cohen, Haitian Sun, R. Alex Hofer, and Matthew Siegler. 2020. Scalable neural methods for reasoning with a symbolic knowledge base. In *International Conference on Learning Representations*.

Catherine Finegan-Dollak, Jonathan K. Kummerfeld, Li Zhang, Karthik Ramanathan, Sesh Sadasivam, Rui Zhang, and Dragomir Radev. 2018. Improving text-to-SQL evaluation methodology. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 351–360, Melbourne, Australia. Association for Computational Linguistics.

Nick McKenna and Mark Steedman. 2022. Smoothing entailment graphs with language models. ArXiv:2208.00318v1 [cs.CL].

Alexander Miller, Adam Fisch, Jesse Dodge, Amir-Hossein Karimi, Antoine Bordes, and Jason Weston. 2016. Key-value memory networks for directly reading documents. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1400–1409, Austin, Texas. Association for Computational Linguistics.

Barlas Oguz, Xilun Chen, Vladimir Karpukhin, Stan Peshterliev, Dmytro Okhonko, Michael Schlichtkrull, Sonal Gupta, Yashar Mehdad, and Scott Yih. 2022. UniK-QA: Unified representations of structured and unstructured knowledge for open-domain question answering. In *Findings of the Association for Computational Linguistics: NAACL 2022*, pages 1535–1546, Seattle, United States. Association for Computational Linguistics.

Srinivas Ravishankar, June Thai, Ibrahim Abdelaziz, Nandana Mihidukulasooriya, Tahira Naseem, Pavan Kapanipathi, Gaetano Rossilleo, and Achille Fokoue. 2021. A two-stage approach towards generalization in knowledge base question answering. *arXiv preprint arXiv:2111.05825*.

Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics.

Subendhu Rongali, Luca Soldaini, Emilio Monti, and Wael Hamza. 2020. Don't parse, generate! a sequence to sequence architecture for task-oriented semantic parsing. In *Proceedings of The Web Conference 2020*, WWW '20, page 2962–2968, New York, NY, USA. Association for Computing Machinery.

Amir Saffari, Armin Oliya, Priyanka Sen, and Tom Ayoola. 2021. End-to-end entity resolution and question answering using differentiable knowledge graphs. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 4193–4200, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Apoorv Saxena, Aditay Tripathi, and Partha Talukdar. 2020. Improving multi-hop question answering over knowledge graphs using knowledge base embeddings. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4498–4507, Online. Association for Computational Linguistics.

Priyanka Sen, Armin Oliya, and Amir Saffari. 2021. Expanding end-to-end question answering on differentiable knowledge graphs with intersection. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 8805–8812, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Jiaxin Shi, Shulin Cao, Lei Hou, Juanzi Li, and Hanwang Zhang. 2021. TransferNet: An effective and transparent framework for multi-hop question answering over relation graph. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 4149–4158, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Haitian Sun, Andrew Arnold, Tania Bedrax Weiss, Fernando Pereira, and William W. Cohen. 2020. Faithful embeddings for knowledge base queries. *Advances in Neural Information Processing Systems*, 33.

Haitian Sun, Tania Bedrax-Weiss, and William Cohen. 2019. PullNet: Open domain question answering with iterative retrieval on knowledge bases and text. 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 2380–2390, Hong Kong, China. Association for Computational Linguistics.

Haitian Sun, Bhuwan Dhingra, Manzil Zaheer, Kathryn Mazaitis, Ruslan Salakhutdinov, and William Cohen. 2018. Open domain question answering using early fusion of knowledge bases and text. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4231–4242, Brussels, Belgium. Association for Computational Linguistics.

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

Peng Wu, Shujian Huang, Rongxiang Weng, Zaixiang Zheng, Jianbing Zhang, Xiaohui Yan, and Jiajun Chen. 2019. Learning representation mapping for relation detection in knowledge base question answering. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 6130–6139, Florence, Italy. Association for Computational Linguistics.

Wen-tau Yih, Matthew Richardson, Chris Meek, Ming-Wei Chang, and Jina Suh. 2016. The value of semantic parse labeling for knowledge base question answering. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 201–206, Berlin, Germany. Association for Computational Linguistics.

Yuyu Zhang, Hanjun Dai, Zornitsa Kozareva, Alexander J Smola, and Le Song. 2018. Variational reasoning for question answering with knowledge graph. In *AAAI*.

A Training Specifications

A.1 Hardware

We performed our experiments on one AWS EC2 instance of p3.2xlarge, which is equipped with one NVIDIA Tesla V100 GPU (16 GiB memory).

A.2 Hyperparameters

We train and test a KG-Flex model constructed to the specifications shown in Table 4.

Hyperparameter	Value
Batch Size	4
Gradient Accumulation	32
Optimizer	Adam
Learning Rate	1e-4
Training Steps	40,000
Relation Embedding Size	768

Table 4: Hyperparameters used in the KG-Flex architecture.

B Added Domains

Our experiments in §4.2 use expert KGs containing specific domains of Freebase. We show summary information for the domains in Table 5.

	Questions	**Relations**	**Triples**
	SimpleQuestions		
Total	108,442	1,830	15,352,572
Film	13,538	61	1,028,076
Medicine	2,881	66	166,886
	WebQuestionsSP		
Total	4,737	585	10,968,596
Film	306	24	814,126
Sports	445	45	176,237
	MetaQA 1-hop		
Total	116,045	9	134,741
directed_by	21,483	1	15,966
written_by	22,193	1	19,543
	MetaQA 2-hop		
Total	148,724	9	134,741
directed_by	51,368	1	15,966
written_by	65,434	1	19,543
	MetaQA 3-hop		
Total	142,744	9	134,741
directed_by	70,227	1	15,966
written_by	60,384	1	19,543

Table 5: Domain summary information for experiments in zero-shot transfer to new domains.