Knowledge Graph-augmented Language Models for Complex Question Answering

Priyanka Sen
Amazon Alexa AI
Cambridge, UK
sepriyan@amazon.com

Sandeep Mavadia
Amazon Alexa AI
Cambridge, UK
smavadia@amazon.com

Amir Saffari
Amazon Alexa AI
Cambridge, UK
amsafari@amazon.com

Abstract

Large language models have shown impressive abilities to reason over input text, however, they are prone to hallucinations. On the other hand, end-to-end knowledge graph question answering (KGQA) models output responses grounded in facts, but they still struggle with complex reasoning, such as comparison or ordinal questions. In this paper, we propose a new method for complex question answering where we combine a knowledge graph retriever based on an end-to-end KGQA model with a language model that reasons over the retrieved facts to return an answer. We observe that augmenting language model prompts with retrieved KG facts improves performance over using a language model alone by an average of 83%. In particular, we see improvements on complex questions requiring count, intersection, or multi-hop reasoning operations.

1 Introduction

Large language models (LMs) have shown great promise in a variety of NLP tasks, including question-answering (QA) (Zhang et al., 2022; Sanh et al., 2022; Wei et al., 2022a). As language models scale, they achieve impressive results on standard QA benchmarks, such as SQuAD (Raffel et al., 2020), and can answer questions either few-shot or zero-shot using only knowledge stored within the model parameters (Roberts et al., 2020). Language models have also been shown to solve complex reasoning tasks by outputting step-by-step instructions from question to answer (Creswell et al., 2022; Wei et al., 2022b). Despite these successes, language models are still prone to hallucinations and can return answers that are incorrect, out-of-date, and not grounded in verified knowledge sources, making them an unsafe choice for a factual question answering service. Additionally, step-by-step reasoning can be computationally expensive as it requires multiple calls to a language model.

Alternatively, knowledge graph-based question answering (KGQA) models (Chakraborty et al., 2019; Fu et al., 2020) are trained to traverse knowledge graph (KG) facts to return answers to questions. These models are faithful and grounded to facts stored in a KG. However KGQA models are often limited in the types of reasoning that they can perform. Most end-to-end KGQA models can perform relation following for single or multiple hops (Cohen et al., 2020), and some models have been trained for set intersection (Sen et al., 2021), union, or difference (Sun et al., 2020), however expanding to general reasoning capabilities remains an open challenge. KGQA models are also restricted to using facts stored in a knowledge graph and can not leverage common world knowledge.

In this paper, we propose a novel method for complex question answering using a KGQA model retriever with a language model reasoner. Our approach harnesses both the ability to traverse over verified facts with a KGQA model, and the ability to reason over text with an LM.

For our KGQA retriever, we train an end-to-end KGQA model based on ReifKB (Cohen et al., 2020) and the Rigel family of models (Saffari et al., 2021; Sen et al., 2021). We use this model to return a weighted set of facts from the knowledge graph that could be useful for answering a given question. We then prompt an LM with the question and the top-k facts retrieved, in a zero-shot setting, and the language model returns a natural language answer. In our experiments over four QA datasets, we show that our approach can outperform using an LM alone by an average of 83%.

2 Related Works

Recent work on using language models for reasoning tasks include Kojima et al. (2023), where the authors prompt the models to output the steps to the answer in addition to the final result. Other methods have also tried to solve questions by break-

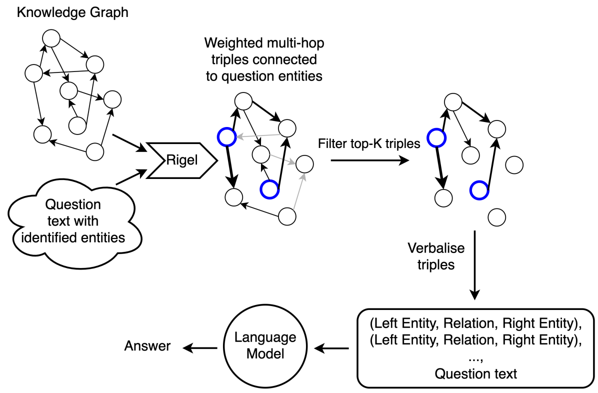
Knowledge Graph
Weighted multi-hop triples connected to question entities
Rigel
Question text with identified entities
Filter top-K triples
Verbalise triples
(Left Entity, Relation, Right Entity),
(Left Entity, Relation, Right Entity),
...,
Question text
Language Model
Answer

Figure 1: Architecture for our model setup. Question entities are shown as blue nodes in the knowledge graph diagrams. See section 3.1 for details.

ing them down into intermediate steps: Wei et al. (2022b) prompts the language model with similar examples where an answer is formed step-wise before providing the requested answer. Creswell et al. (2022) fine-tune several models with the task of choosing relevant knowledge until a satisfactory answer is reached. Although these methods improve language model performance, it can be costly to fine-tune a large language model or to pass an input through a language model multiple times, especially at runtime. In this work, we instead build a lighter weight retriever model to collect relevant facts followed by a single call to a language model.

Different data sources have also been used for retrieval: Lazaridou et al. (2022) uses Google search. Kang et al. (2022) also retrieves facts from a knowledge graph but requires fine-tuning a language model, which can be expensive for the larger models. Recently, Baek et al. (2023) used a similarity metric between KG facts and questions to retrieve relevant facts to add to the prompt of a language model. However Baek et al. (2023) found that similarity alone is not always enough to find relevant facts for complex question. In this work we present a more sophisticated model for identifying relevant facts with a KGQA model.

3 Method

We propose a new method for question answering using a KGQA model to retrieve facts from a knowledge graph, and a language model to reason over the question and facts to return an answer.

3.1 Model description

As shown in Figure 1, our model has three main components:

1. A sequence-to-sequence KGQA model (which we refer to as RIGEL based on Saffari et al. (2021); Sen et al. (2021)) for predicting a distribution over which relations to follow in a knowledge graph.
2. A differentiable knowledge graph (DKG), where the KG is stored in three linear maps from left-entities, relations, and right-entities to triples respectively, represented as sparse binary matrices (Cohen et al., 2020).
3. A language model for interpreting the questions and reasoning over facts provided from the KG by the two previous components.

We do not yet integrate entity resolution, so question entities are provided from the datasets.

There are three steps to running the model in inference. First, Rigel is used to estimate a distribution over relations for each hop using a sequence-to-sequence architecture. We initialize the encoder using RoBERTa-base (Liu et al., 2019). The decoder predicts a distribution over relations in the knowledge graph. This decoding step is performed for up to M hops (in our experiments, $M = 2$).

Second, the question entities and the distribution over relation are used to extract weighted triples from the knowledge graph. We represent the question entities as a one-hot vector in entity-space and map this to a vector of triples using the left-entity to triple sparse matrix in the DKG. Similarly, for relations, we use the vector over relations predicted by the Rigel model and map this to a vector of triples using the relation to triple matrix in the DKG. Finally we take the Hadamard (element-wise) product to extract a weighted vector of triples. For the second hop, we map the triple vector back to entities using the right-entity to triple matrix in the DKG and repeat the process above.

We retain only the top-k triples for each hop (in our experiments $k = 10$) and convert them into natural language using the names of the entities and relations as stored within the KG. We also include inverse triples within our DKG where the relations are prefixed with "<inv>-" when converted to natural language, e.g. "(Paris, <inv>-capital-of, France)". We also include literal values for numbers, strings and dates in our DKG as right entities.

Finally, we run inference in a zero-shot setting with a pretrained language model. We compose a prompt following the template: "*Given the following context:* "{context}" *Answer the question:*

{question}. *Answer:*" where the context is the set of filtered triples formatted as "*(left entity, relation, right entity), . . . , (left entity, relation, right entity)*". We input this prompt into a language model to output an answer.

3.2 Training

Of the three components outlined in section 3.1, we only train the Rigel KGQA model. The DKG is instantiated from a static dump of the Wikidata (Vrandečić and Krötzsch, 2014) knowledge graph, and the language models in our experiments are not fine-tuned.

Rigel is trained using the train and dev sets of KGQA datasets annotated with natural language questions, question entities, and answer entities. As described in section 3.1 we estimate a distribution over entities for each hop. During training, we also jointly learn an attention mechanism which is conditioned on the question embedding to predict how much to weigh answer entities returned for each hop. We use a binary cross-entropy objective to allow for multiple answer entities. For more information on training the Rigel model see Sen et al. (2021). We leave end-to-end training of both the KGQA model and the LM to future work.

4 Experimental Setup

4.1 Datasets

We use four KGQA datasets in our experiments with Wikidata as the knowledge graph. For datasets built using FreeBase, we link entities to Wikidata using the *FreeBase ID* Wikidata property. The train / dev sets for each dataset are used to train the Rigel model, and all results are reported on the test sets.

- **WebQuestions** (Berant et al., 2013) is an English question-answering dataset of 4,737 questions (2,792 train, 306 dev, 1,639 test) originally built on FreeBase. WebQuestions includes questions requiring multiple hops and set intersections.

- **ComplexWebQuestions** (Talmor and Berant, 2018) is an extended version of WebQuestions with 34,689 questions (27,649 train, 3,509 dev: since the test set is not public, we use the dev set for testing) in English requiring complex operations, such as multiple hops and temporal constraints.

- **Mintaka** (Sen et al., 2022) is a complex question answering dataset of 20,000 questions (14,000 train, 2,000 dev, 4,000 test) linked to Wikidata and using complex operations such as comparisons and set operations. We use the English subset.

- **LC-QuAD** is a dataset of 30,000 synthetically generated English questions and SPARQL parses. We use the subset with SPARQL parses that return a valid answer from Wikidata (20,438 train, 5,230 test). These questions include complex operations such as multi-hop and count.

4.2 Language Models

We evaluate our method using language models from four families of models:

- **Flan-T5** (Chung et al., 2022) models are an extension of T5 encoder-decoder models that have been *instruction tuned* on a large set of instructions that were automatically generated using existing datasets and templates. We use the Flan-T5 Small (80M parameters), XL (3B), and XXL (11B) models.

- **T0** (Sanh et al., 2022) models are encoder-decoder models that are trained on a variety of *prompts*, which are automatically built from supervised datasets using templates. We use the T0 (11B) and T0 3B (3B) models.

- **OPT** (Zhang et al., 2022) models are large, open-source, decoder-only models that have been trained to roughly match the performance of GPT-3 models. We use the 13B parameter version of OPT.

- **AlexaTM** (Soltan et al., 2022) is a 20 billion parameter encoder-decoder model trained on publicly available data in multiple languages.

4.3 Training Specifications

We train each of our Rigel models on a single NVIDIA Tesla V100 GPU for 40,000 steps. We run inference over our language models using four Tesla V100 GPUs and distribute across GPUs using Hugging Face Accelerate (Gugger et al., 2022).

4.4 Evaluation metric

Our datasets provide answer entities (e.g., Wikidata Q-codes). To evaluate the performance of the LM's natural language output, we test if any of the provided answer entity names or their aliases as

			Flan-T5			**T0**			
Dataset	**Experiment**	**Rigel**	**S**	**XL**	**XXL**	**3B**	**11B**	**OPT**	**ATM**
WebQ	No Knowledge	–	16.29	40.15	45.15	29.10	34.05	26.85	38.56
	Random Facts	–	21.90	28.49	39.96	28.07	36.30	48.02	41.98
	Rigel Facts	48.9	**45.52**	**55.58**	**59.79**	**53.33**	**55.64**	**57.60**	**55.40**
	% improvement	–	179%	38%	32%	83%	63%	115%	44%
CWQ	No Knowledge	–	9.63	28.79	31.00	20.26	24.98	18.19	27.20
	Random Facts	–	14.69	23.96	31.15	20.86	26.85	28.13	29.41
	Rigel Facts	29.21	**25.55**	**36.09**	**40.38**	**32.54**	**36.35**	**32.54**	**35.72**
	% improvement	–	165%	25%	30%	61%	46%	79%	31%
Mintaka	No Knowledge	–	12.65	24.63	30.15	21.13	30.08	38.53	28.48
	Random Facts	–	12.20	20.98	33.63	21.33	30.40	**42.20**	33.00
	Rigel Facts	21.7	**20.58**	**33.50**	**37.90**	**29.28**	**33.35**	40.03	**35.60**
	% improvement	–	63%	36%	26%	39%	11%	4%	25%
LC-QuAD	No Knowledge	–	3.90	8.15	3.82	8.32	9.31	8.75	11.79
	Random Facts	–	8.40	8.91	**11.24**	9.71	10.65	12.65	12.81
	Rigel Facts	27.86	**15.65**	**22.82**	9.41	**20.54**	**22.32**	**20.93**	**22.30**
	% improvement	–	301%	180%	146%	147%	140%	139%	89%

Table 1: Results by language model and dataset over two baselines (No Knowledge and Random Facts) and our proposed method, Rigel Facts. % improvement shows the percentage improvement over No Knowledge to Rigel Facts. Rigel shows the baseline of using Rigel alone with no language model.

stored in Wikidata exist within the LM output. We also lower case text in the prediction and remove punctuation, articles, and extra white space.

5 Results

We evaluate our method on four complex question-answering datasets using seven language models. We compare against two baselines.

- **No Knowledge**: we provide the question with no additional context. The prompt is "*Question:* {question} *Answer:*".
- **Random Facts**: we provide k random facts (k = 10) sampled uniformly over all facts reachable in one hop from the question entities.

The results are reported in Table 1, with the % improvement showing the percentage of improvement from the No Knowledge baseline to our proposed Rigel Facts method. We also report scores for the Rigel model alone in the Rigel column.

These results show that in almost all cases, a language model using facts retrieved from our Rigel model outperforms No Knowledge and Random Facts. For smaller models such as Flan-T5, using Rigel facts improves performance by up to 300%. Larger models, such as AlexaTM, start with higher baselines using no knowledge, but still see an average of 47% improvement across datasets. Exceptions are OPT on Mintaka and Flan-T5 XXL on LC-QuAD, where random facts outperform Rigel. We observe that in many of the questions where augmenting with random facts performs better than Rigel facts, neither provide useful information. Interestingly, however, random facts still encourage the LM to output the correct answer.

The No Knowledge results show that models can answer questions without additional facts. Larger models with no facts can even outperform smaller models with Rigel facts, for example, AlexaTM vs. Flan-T5 on Mintaka (28.48 vs. 20.58). Nevertheless, it is promising to see smaller models become more competitive with the help of a retriever.

The use of random facts shows mixed results. Random facts rarely outperform Rigel, but compared to the No Knowledge baseline, random facts can sometimes help, as seen across models on ComplexWebQuestions and LC-QuAD. In other cases,

		Flan-T5			T0				
Question Type	**Experiment**	**S**	**XL**	**XXL**	**3B**	**11B**	**OPT**	**ATM**	**Average**
Comparative	No Facts	**48.50**	63.00	64.00	**49.25**	**59.75**	58.50	57.25	**56.75**
	Random Facts	36.00	61.25	62.50	44.50	57.75	**60.00**	**63.00**	55.00
	Rigel Facts	42.75	**64.25**	**65.50**	42.00	55.50	54.75	60.00	55.12
Count	No Facts	16.75	26.25	**33.00**	21.25	25.00	25.00	40.75	27.56
	Random Facts	17.25	17.75	28.50	27.25	28.75	**51.00**	43.50	30.57
	Rigel Facts	**23.75**	**27.25**	31.25	**40.75**	**32.00**	49.00	**48.75**	**36.47**
Difference	No Facts	4.25	16.75	19.50	**17.00**	**21.00**	20.00	**28.25**	19.28
	Random Facts	6.75	11.75	20.50	11.25	15.50	**29.00**	21.75	16.64
	Rigel Facts	**15.50**	**17.50**	**24.00**	14.25	17.00	28.75	27.25	**20.44**
Generic	No Facts	2.12	16.50	24.25	18.12	28.75	48.38	37.50	27.12
	Random Facts	11.62	20.50	35.75	19.62	30.88	**50.00**	43.62	30.29
	Rigel Facts	**20.50**	**34.25**	**41.12**	**30.88**	**36.75**	47.12	**45.50**	**37.61**
Intersection	No Facts	1.75	20.00	28.50	22.50	35.25	**54.00**	42.50	31.47
	Random Facts	8.50	22.50	37.00	18.00	31.25	51.00	40.50	29.82
	Rigel Facts	**16.00**	**40.25**	**44.75**	**35.00**	**41.00**	49.50	**45.75**	**39.81**
Multi-hop	No Facts	3.00	7.25	12.75	8.25	13.25	20.00	13.25	12.44
	Random Facts	2.75	9.50	18.25	6.50	14.25	24.00	15.75	13.00
	Rigel Facts	**13.50**	**22.25**	**27.75**	**20.50**	**21.25**	**25.75**	**21.00**	**22.69**
Ordinal	No Facts	1.50	9.50	16.50	10.00	15.50	27.75	20.25	15.44
	Random Facts	6.75	9.50	17.75	9.50	**18.00**	**29.75**	20.75	16.00
	Rigel Facts	**12.00**	**16.50**	**23.75**	**17.25**	**18.00**	24.25	**24.25**	**19.84**
Superlative	No Facts	1.25	11.75	16.00	16.00	**19.25**	**28.75**	21.50	17.12
	Random Facts	6.00	12.00	**21.75**	12.25	17.50	28.25	23.75	17.36
	Rigel Facts	**10.50**	**18.75**	**21.75**	**16.75**	14.00	25.00	**24.00**	**19.34**
Yes/No	No Facts	**45.25**	59.00	**62.75**	**49.00**	**63.25**	**54.00**	37.00	**53.16**
	Random Facts	14.75	24.50	58.25	44.50	60.50	49.25	**51.25**	43.29
	Rigel Facts	30.75	**59.75**	58.50	44.50	62.50	49.25	**51.25**	52.12
Average	No Facts	13.82	25.56	30.81	23.49	31.22	37.38	33.14	28.93
	Random Facts	12.26	21.03	33.36	21.49	30.49	**41.36**	35.99	28.00
	Rigel Facts	**20.58**	**33.42**	**37.60**	**29.10**	**33.11**	39.26	**38.64**	**33.72**

Table 2: A breakdown of results by the different complexity types in the Mintaka dataset

random facts can hurt performance, as seen in Flan-T5 XL on WebQuestions (from 40.15 to 28.49) and Mintaka (from 24.63 to 20.98). This can be attributed to random facts adding in distractors that some models are more susceptible to. For example, given the QA pair "*Q: Where does Princess Leia live? A: Alderaan*", if the random facts include *"Leia Organa place of birth Polis Massa*", the model can incorrectly answer Polis Massa.

We also show results in Table 2 as a breakdown of performance by complexity type on the Mintaka dataset. On average, we see that Rigel facts help across complexity types. The highest gains are in Count, Intersection, and Multi-hop questions. These are also the areas that a model like Rigel, which traverses a knowledge graph by following relations, is best suited for. Finding facts for comparatives or yes/no questions are less reliable since the training signal can be weak and there can be multiple paths that spuriously lead to the correct answer. For example, to answer *Who is older, The Weeknd or Drake?*, there are several ways to get to

Question

How many countries were in the Central Powers alliance in World War I?

Random Facts	Rigel Facts
· Central Powers has part Austria-Hungary	· Central Powers has part Ottoman Empire
· Central Powers Commons category Central Powers	· Central Powers has part Kingdom of Bulgaria
· Central Powers participant in World War I	· Central Powers has part German Empire
· Central Powers instance of military alliance	· Central Powers has part Austria-Hungary

Predictions	Model	No Knowledge	Random Facts	Rigel Facts
	Flan-T5 XXL	6 ✗	2 ✗	4 ✓

Question

Where did the author of Pet Sematary go to college?

Random Facts	Rigel Facts
· Pet Sematary author Stephen King	· Stephen King education Lisbon High School
· Pet Sematary follows Christine	· Stephen King education University of Maine
· Pet Sematary language of work or name English	· Pet Sematary author Stephen King
· Pet Sematary publisher Doubleday	· Pet Sematary notable work Stephen King

Predictions	Model	No Knowledge	Random Facts	Rigel Facts
	T0	University of Michigan ✗	Dartmouth College ✗	University of Maine ✓

Question

What was the first book in the Lord of the Ring's series?

Random Facts	Rigel Facts
· Lord of the Rings characters Gandalf	· Fellowship of the Ring follows The Hobbit
· Lord of the Rings characters Elrond	· Two Towers follows Fellowship of the Ring
· Lord of the Rings translator Maria Skibniewska	· Return of the King follows Two Towers
· Lord of the Rings nominated for Prometheus Award	· Appendices follows Return of the King

Predictions	Model	No Knowledge	Random Facts	Rigel Facts
	T0	Fellowship of the Ring ✓	Fellowship of the Ring ✓	The Hobbit ✗

Table 3: Examples of questions and model predictions. For simplicity, we only show the top four facts. In **Predictions**, No Knowledge is only given the question. Random and Rigel Facts are given the question and the respective facts. The correct answer is indicated with a ✓. Incorrect answers are indicated with a ✗.

the answer entity *Drake* without following a *date of birth* relation and performing a comparison. In future work, we plan to explore different ways to train the Rigel model to provide a better training signal of which facts will be useful to the LM.

Finally, in Table 3, we provide examples. The first example is a *count* question, where the LM seems to count the entities Rigel returns get to the correct answer. The second example is of a *multi-hop* question. Of note is that Rigel's top fact is about a high school, but the LM is able to recover and return a college instead. The third example is an *ordinal* question. Since the Rigel facts do not specify which books are part of the series, the model returns an incorrect answer by trying to stay faithful to the facts given. Relying on facts given rather than facts in the parameters can be a desirable trait for an LM, however this example highlights that more work needs to be done on improving the KGQA retriever.

6 Conclusion

In this paper, we show how facts from a KGQA based retriever can be combined with a language model to help answer complex questions. Our results show improvements over calling a language model directly over four datasets, and in particular on complexity types such as multi-hop and count questions. We present our method as a promising way to leverage a knowledge graph, which contains verified and up-to-date facts, with a single call to a language model. In future work, we plan to improve performance across more complexity types and aim to explore ways to update the training of our KGQA retriever with feedback from the language model.

7 Limitations

We present a method for question answering using a KGQA retriever and a language model reasoner. Limitations of our method include a lack of an integrated entity resolution system when training our KGQA model: we instead rely on annotated entities from the datasets. While our KGQA architecture is robust to new entities added at test time, it does require retraining when new relations are added to the KG or if a different target KG is used. Additionally, our results are based on training and evaluating on one dataset at a time; training on a mix of datasets could lead to better generalization, however this is not tested.

References

Jinheon Baek, Alham Fikri Aji, and Amir Saffari. 2023. Knowledge-augmented language model prompting with knowledge graphs for zero-shot question answering. In *Proceedings of the First Workshop on Matching*, Toronto, Canada. Association for Computational Linguistics.

Jonathan Berant, Andrew Chou, Roy Frostig, and Percy Liang. 2013. Semantic parsing on freebase from question-answer pairs. In *Proceedings of the 2013 conference on empirical methods in natural language processing*, pages 1533–1544.

Nilesh Chakraborty, Denis Lukovnikov, Gaurav Maheshwari, Priyansh Trivedi, Jens Lehmann, and Asja Fischer. 2019. Introduction to neural network based approaches for question answering over knowledge graphs. *arXiv preprint arXiv:1907.09361*.

Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Eric Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, et al. 2022. Scaling instruction-finetuned language models. *arXiv preprint arXiv:2210.11416*.

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*.

Antonia Creswell, Murray Shanahan, and Irina Higgins. 2022. Selection-inference: Exploiting large language models for interpretable logical reasoning. *arXiv preprint arXiv:2205.09712*.

Bin Fu, Yunqi Qiu, Chengguang Tang, Yang Li, Haiyang Yu, and Jian Sun. 2020. A survey on complex question answering over knowledge base: Recent advances and challenges. *arXiv preprint arXiv:2007.13069*.

Sylvain Gugger, Lysandre Debut, Thomas Wolf, Philipp Schmid, Zachary Mueller, and Sourab Mangrulkar. 2022. Accelerate: Training and inference at scale made simple, efficient and adaptable. `https://github.com/huggingface/accelerate`.

Minki Kang, Jin Myung Kwak, Jinheon Baek, and Sung Ju Hwang. 2022. Knowledge-consistent dialogue generation with knowledge graphs. In *ICML 2022 Workshop on Knowledge Retrieval and Language Models*.

Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. 2023. Large language models are zero-shot reasoners. *arXiv preprint arXiv:2205.11916*.

Angeliki Lazaridou, Elena Gribovskaya, Wojciech Stokowiec, and Nikolai Grigorev. 2022. Internet-augmented language models through few-shot prompting for open-domain question answering. *arXiv preprint arXiv:2203.05115*.

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 preprint arXiv:1907.11692*.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *The Journal of Machine Learning Research*, 21(1):5485–5551.

Adam Roberts, Colin Raffel, and Noam Shazeer. 2020. How much knowledge can you pack into the parameters of a language model? In *Empirical Methods in Natural Language Processing (EMNLP)*.

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.

Victor Sanh, Albert Webson, Colin Raffel, Stephen Bach, Lintang Sutawika, Zaid Alyafeai, Antoine Chaffin, Arnaud Stiegler, Arun Raja, Manan Dey, M Saiful Bari, Canwen Xu, Urmish Thakker, Shanya Sharma Sharma, Eliza Szczechla, Taewoon Kim, Gunjan Chhablani, Nihal Nayak, Debajyoti Datta, Jonathan Chang, Mike Tian-Jian Jiang, Han Wang, Matteo Manica, Sheng Shen, Zheng Xin Yong, Harshit Pandey, Rachel Bawden, Thomas Wang, Trishala Neeraj, Jos Rozen, Abheesht Sharma, Andrea Santilli, Thibault Fevry, Jason Alan Fries, Ryan

Teehan, Teven Le Scao, Stella Biderman, Leo Gao, Thomas Wolf, and Alexander M Rush. 2022. Multitask prompted training enables zero-shot task generalization. In *International Conference on Learning Representations*.

Priyanka Sen, Alham Fikri Aji, and Amir Saffari. 2022. Mintaka: A complex, natural, and multilingual dataset for end-to-end question answering. In *Proceedings of the 29th International Conference on Computational Linguistics*, pages 1604–1619, Gyeongju, Republic of Korea. International Committee on 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.

Saleh Soltan, Shankar Ananthakrishnan, Jack FitzGerald, Rahul Gupta, Wael Hamza, Haidar Khan, Charith Peris, Stephen Rawls, Andy Rosenbaum, Anna Rumshisky, et al. 2022. Alexatm 20b: Few-shot learning using a large-scale multilingual seq2seq model. *arXiv preprint arXiv:2208.01448*.

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.

Alon Talmor and Jonathan Berant. 2018. The web as a knowledge-base for answering complex questions. *arXiv preprint arXiv:1803.06643*.

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

Jason Wei, Maarten Bosma, Vincent Y. Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M. Dai, and Quoc V. Le. 2022a. Finetuned language models are zero-shot learners. In *ICLR*.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Ed H. Chi, Quoc Le, and Denny Zhou. 2022b. Chain of thought prompting elicits reasoning in large language models. *arXiv preprint arXiv:2201.11903*.

Susan Zhang, Stephen Roller, Naman Goyal, Mikel Artetxe, Moya Chen, Shuohui Chen, Christopher Dewan, Mona Diab, Xian Li, Xi Victoria Lin, Todor Mihaylov, Myle Ott, Sam Shleifer, Kurt Shuster, Daniel Simig, Punit Singh Koura, Anjali Sridhar, Tianlu Wang, and Luke Zettlemoyer. 2022. OPT: open pre-trained transformer language models. *arXiv preprint arXiv:2205.01068*.