SKILL: Structured Knowledge Infusion for Large Language Models

Fedor Moiseev* [1,2] **Zhe Dong**† [2] **Enrique Alfonseca** [2] **Martin Jaggi** [1]
[1]EPFL, Switzerland [2]Google, Switzerland
{femoiseev, zhedong, ealfonseca}@google.com, martin.jaggi@epfl.ch

Abstract

Large language models (LLMs) have demonstrated human-level performance on a vast spectrum of natural language tasks. However, it is largely unexplored whether they can better internalize knowledge from a structured data, such as a knowledge graph, or from text. In this work, we propose a method to infuse structured knowledge into LLMs, by directly training T5 models on factual triples of knowledge graphs (KGs). We show that models pre-trained on Wikidata KG with our method outperform the T5 baselines on FreebaseQA and WikiHop, as well as the Wikidata-answerable subset of TriviaQA and NaturalQuestions. The models pre-trained on factual triples compare competitively with the ones on natural language sentences that contain the same knowledge. Trained on a smaller size KG, WikiMovies, we saw 3× improvement of exact match score on MetaQA task compared to T5 baseline. The proposed method has an advantage that no alignment between the knowledge graph and text corpus is required in curating training data. This makes our method particularly useful when working with industry-scale knowledge graphs.

1 Introduction

Large pre-trained language models, such as BERT (Devlin et al., 2019), GPT-3 (Brown et al., 2020), T5 (Raffel et al., 2019), REALM (Guu et al., 2020) and ERNIE (Sun et al., 2021) have become the state-of-the-art technology for many tasks. They are commonly pre-trained using unstructured text corpora, on tasks such as next word prediction, next sentence prediction (NSP) or masked language modelling (MLM). Especially for T5, self-supervised learning on unlabelled text corpus with MLM has been a common pre-training recipe (Roberts et al., 2020). This is normally followed by a fine-tuning step on the task of interest (Ruder et al., 2019), although large language models have also proved useful without this task-specific fine-tuning (Brown et al., 2020).

Beyond the capacity of contextual understanding, human-level language understanding pivots on the knowledge about the world. The world knowledge is often expressed as factual triples (c.f. Ji et al., 2020), in the form of (*subject entity*, *relation*, *object entity*). A knowledge graph (KG) defined by a set of factual triples consists of the subjects and objects as vertices/nodes, and the relations forming the edges connecting them. Most of the large scale KGs (e.g. Wikidata, Vrandečić and Krötzsch, 2014) are stored in triple format.

LLMs demonstrate some capacity of learning world knowledge from the natural text corpus (Roberts et al., 2020), but it is unclear to what degree they are also able to learn and memorize new knowledge directly from structured KG triples, or from text describing them explicitly.

In order to infuse knowledge into a LLM, one option is to generate a textual version of the knowledge base, and apply the standard training objectives, e.g. MLM. This is unfortunately highly nontrivial. One can either align sentences with KG triples, as done in ERNIE (Sun et al., 2021), or generate sentences from triples, as done in KELM (Agarwal et al., 2021). These approaches are unfortunately hard to port to knowledge graphs with different schemas. These processes are also lossy in that not every triple can be aligned or produce a valid sentence, and there is not a good understanding whether this can introduce unnecessary selection biases on top of biases existing in the original KG.

In this work, we propose a method of **Knowledge Infusion for Large Language Models (SKILL)**, where LLMs directly learns from knowledge triples. Experiment results shows the checkpoints trained with proposed method on Wikidata KG outperform the T5 baselines on four standard

*Work done during internship at Google.
†Correspondence Author.

Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pages 1581 - 1588
July 10-15, 2022 ©2022 Association for Computational Linguistics

closed-book question-answering (QA) tasks. With a smaller KG, WikiMovies, the proposed method gain 3× exact match score performance improvement on MetaQA task. The models learning directly from knowledge triples performs competitively with the ones with the aligned natural sentences that contain the same amount of knowledge. Being able to learn directly from knowledge triples enables easy addition of structured knowledge into language modeling pre-training.

2 Related work

Previous works that use knowledge graphs to enhance the quality of knowledge-intensive downstream tasks can be divided into two groups: using knowledge graphs at the inference time, and infusing knowledge into the model weights at the pre-training time. The proposed method falls in the latter group.

Explicit usage of knowledge graphs. A retrieval-augmented model is commonly used, in order to retrieve and apply the knowledge from external memories or sources. FILM (Verga et al., 2021) and EaE (Févry et al., 2020) extend Transformer (Vaswani et al., 2017) models with external entity (both FILM and EaE) and fact (FILM) memories. REALM (Guu et al., 2020) is pre-trained to perform reasoning over a large textual knowledge corpus on-the-fly during inference. UniK-QA (Oguz et al., 2020) combines the structured and unstructured information to improve the open-domain QA tasks with a retriever-reader framework. The main difference between the proposed method, SKILL, and retrieval-augmented models is that SKILL doesn't introduce retrieval system or external memories to the model, but it directly embeds knowledge into the model parameters, which introduces no extra cost at inference time.

Knowledge infusion. A common way of parameterized knowledge infusion is to map or convert structured knowledges into natural language text. ERNIE 3.0 (Sun et al., 2021) trains a knowledge-enhanced model on a corpus combining triples and their aligned sentences, by randomly masking relation in a triple or words in a sentence. On the contrary, SKILL trains only on triples.

KnowBert (Peters et al., 2019) incorporates knowledge from Wikipedia and WordNet (Miller, 1995) into a BERT model through entity embeddings with knowledge-attention and re-contextualization mechanism. BERT-MK (He et al., 2020) is a BERT-based model that integrates graph contextual knowledge of a medical KG, which demonstrates the utility of graph-level knowledge. These approaches requires entity linking and sentences contextualizing the knowledge graph information.

KG-FiD (Yu et al., 2021) extends the Fusion-in-Decoder model (Izacard and Grave, 2021) with a module that filters and re-ranks passages based on structural connections in knowledge graph between entities described in those passages. In contrast to the SKILL method that we propose, it requires the existence of natural text passages describing each knowledge graph entity, so Wikipedia corpus was used since it naturally provides articles that describe entities.

Heinzerling and Inui (2021) explored the ability of language models to memorize and understand information from knowledge graphs, but used natural language representation of triples based on predefined templates instead of structured representation. Usage of predefined templates significantly limits scalability and therefore only relatively small knowledge graphs were used, such as Google-RE[1].

In contrast to the new method presented in this paper, all of these approaches require an explicit mapping between the knowledge graph entities or facts and corresponding natural language sentences, which can limit applications to industry-scale knowledge graphs that don't have such a mapping.

Different goals of using knowledge graphs. Besides that, some papers embed knowledge into model weights but pursue different goals rather than improving performance on downstream tasks. COMET (Bosselut et al., 2019) is most similar to our work and trains a commonsense-aware Transformer Language Model by learning to generate loosely structured commonsense descriptions in the natural language given the structured knowledge. Similar to us, it also uses KG triples in surface form as a source for training data, but in contrast to our research, the final goal of COMET is to generate new knowledge instead of utilizing existing ones. Another important difference is the scale: COMET uses Atomic (Sap et al., 2019) and ConceptNet (Speer et al., 2017) Knowledge Graphs

[1] https://ai.googleblog.com/2013/04/50000-lessons-on-how-to-read-relation.html

that are much smaller than Wikidata (Vrandečić and Krötzsch, 2014).

KELM (Agarwal et al., 2021) fine-tunes a T5 model to convert KGs to synthetic natural language sentences to augment existing pre-training corpora. We build our research on top of it and use the KELM dataset to compare structured and natural language representations of knowledge.

3 Method

There are two components of knowledge infusion for LLMs (SKILL): the corpus and the training method. We introduce the method based on Wikidata KG, but it can be applied to any other KGs.

Training corpus. We use two corpora with different knowledge representations: Wikidata KG (Vrandečić and Krötzsch, 2014) in triple format, and KELM corpus[2] (Agarwal et al., 2021) as synthetic natural language sentences converted from Wikidata KG. The KELM corpus contains $15,628,486$ synthetic sentences. To ensure two corpora share the same knowledge, we take the snapshot of the Wikidata KG used to created the KELM corpus, which contains $35,697,715$ triples.

To prevent the degradation of model performance on natural language understanding, we mix the Wikidata corpus or KELM corpus with natural text from C4 (Raffel et al., 2019), $50:50$, for the knowledge infusion training data.

Training method. T5 (Raffel et al., 2019) was trained through masked-language modelling with random span corruption on the C4 corpus. Roberts et al. (2020) found that masking salient terms (Guu et al., 2020) in pre-training T5 models, instead of masking random token spans, could significantly improve the performance on downstream tasks, e.g. closed-book QA.

We apply salient span masking for unsupervised learning in our knowledge-infusing training. To mask the same amount of information is for both corpora, the following method is applied. For a knowledge triple, we mask either the subject or object entity. For a KELM sentence, we identify the aligned triple, with details in Appendix A, and mask the full spans corresponding to the subject or object in the triple. The *relation* tokens are never masked, as there is no robust way to map the abstract relation in knowledge triples to natural language tokens in KELM sentences. Examples of the inputs for both corpora are in Table 1.

4 Experiments

We assess SKILL by training and evaluating the knowledge infused models on closed-book QA tasks, where questions are provided without supporting context and external knowledge.

4.1 Experiment Setup

SKILL pre-training. We apply SKILL on three T5.1.1 pre-trained checkpoints[3], base, large, and XXL, with sizes of ~ 250M, ~ 800M and ~ 11B parameters, respectively. For T5.1.1-base and -large, SKILL training is performed for 500K steps with batch size 1024, which translates to ~ 7.17 epochs on Wikidata KG and ~ 16.38 epochs in KELM sentences. For T5.1.1-XXL, the model is trained for 100K steps to finish training in a feasible time.

As baseline we use pre-trained T5 checkpoints of the same size. To make sure that improvements come from knowledge infusion instead of from longer C4 pre-training, we use a second baseline by further training the T5 checkpoints on C4 for half of the aforementioned steps, to match the amount of C4 pre-training used in SKILL.

All the model variations are optimized by AdaFactor (Shazeer and Stern, 2018) with 10^{-3} learning rate and 0.1 dropout rate, the same settings that were used for T5.

Fine-tuning on closed-book QA tasks. We evaluate the checkpoints by fine-tuning on the following QA benchmarks: FreebaseQA (Jiang et al., 2019), WikiHop (Welbl et al., 2018), TriviaQA (Joshi et al., 2017) and NaturalQuestions (Kwiatkowski et al., 2019), with the aforementioned hyper-parameters for optimization and 128 batch size. For the benchmarks without a *test* split, we use the *dev* split for test, and the last 10% of *train* as *dev* split.

The Exact Match (EM) scores on the test sets are calculated after being fine-tuned for 50K steps for T5.1.1-base and -large models, and 10K steps for -XXL models. All models converged with no noticeable over-fitting according to the EM scores on validation sets.

Wikidata-answerable QA. We found that the majority of the questions in FreebaseQA and Wiki-

[2] Data is available at https://github.com/google-research-datasets/KELM-corpus

[3] https://goo.gle/t5-checkpoints

Wikidata triple	KELM sentence	Wikidata input	KELM input	Target
("Pulp Fiction", "award received", "Palme d'Or")	Quentin Tarantino won the Palme d'Or in 1994 for Pulp Fiction.	Pulp Fiction, award received, [MASK]	Quentin Tarantino won the [MASK] in 1994 for Pulp Fiction.	Palme d'Or

Table 1: Example inputs for SKILL pre-training with Wikidata and KELM corpora.

Model	FreebaseQA		WikiHop		TQA-matched		TQA		NQ-matched		NQ	
	dev	test	dev	test	dev	test	dev	test	dev	test	dev	test
base	25.24	27.55	19.09	18.38	31.24	33.55	22.64	22.93	36.64	32.68	25.04	25.48
base + C4	26.19	28.33	19.57	19.36	32.9	34.4	24.54	25.39	36.98	32.03	**25.88**	25.84
base + WikiKG	**26.92**	**28.38**	20.28	**20.22**	**34.21**	35.08	24.73	**25.77**	**37.41**	**33.33**	25.51	25.76
base + KELM	26.64	28.15	**20.62**	19.81	33.64	**35.54**	**25.22**	25.75	36.98	32.9	25.31	**26.2**
large	30.22	32.88	20.92	21.12	36.7	38.09	29.24	30.03	39.22	35.06	27.12	27.15
large + C4	32.55	34.01	22.5	21.51	38.78	40.6	30.32	30.83	39.74	35.5	27.46	28.17
large + WikiKG	**33.22**	**35.29**	**23.5**	**23.4**	39.19	**41.02**	29.74	30.47	**41.12**	**35.93**	27.38	27.89
large + KELM	32.65	34.16	23.34	22.91	**39.45**	40.76	**30.51**	**30.65**	40.95	35.5	**27.67**	**28.56**
XXL	43.67	45.02	24.76	24.8	51.73	53.1	42.44	42.21	46.47	43.72	31	32.27
XXL + C4	42.01	44.14	23.34	22.23	50.59	52.19	40.66	40.99	45.43	40.26	30.35	31.08
XXL + WikiKG	45.22	**47.25**	**27.57**	**27.65**	**54.17**	54.18	42.55	**43.54**	**49.14**	**44.37**	31.11	**32.74**
XXL + KELM	**45.42**	45.9	26.11	26.26	53.65	**54.21**	**42.68**	42.95	48.53	44.16	**31.79**	32.6

Table 2: Exact match scores achieved by fine-tuning the checkpoints on closed-book QA tasks. base, large, XXL represent the corresponding T5.1.1-* checkpoints. ⋆-C4 are the checkpoints additionally trained on C4 corpus as discussed in Section 3. ⋆-WikiKG and ⋆-KELM are the checkpoints trained on Wikidata KG triple corpus and KELM sentence corpus. The best performed checkpoints are in bold. Details about datasets are in Appendix B.

Hop can be answered directly from triples in Wikidata. This is because FreebaseQA was created by matching question-answer pairs with triples in Freebase (Bollacker et al., 2008), most of which was imported into Wikidata (Vrandečić and Krötzsch, 2014). For WikiHop, the questions were generated from Wikidata triples.

However, TriviaQA and NaturalQuestions were created independently of Wikidata, and not every question can be answered using this knowledge base. We found frequent freshness issues, e.g. the golden answer for question "Who is the largest supermarket chain in the UK?" is "Aldi", while today it would be "Tesco". Some other questions can not be answered by WikiData, e.g. "Who, during a radio microphone test in 1984 said, 'I just signed legislation which outlaws Russia forever. The bombing begins in five minutes?'", with the golden answer "Ronald Reagan".

To mitigate this, we created subsets of TriviaQA (TQA) and NaturalQuestions (NQ) that were somewhat more likely to have answers in Wikidata. We selected all the items for which there exist a triple in Wikidata that has the answer either as subject or object, and the other entity in the triple is mentioned in the question. We match the entities by entity name, case-insensitive. We name the Wikidata-aligned version of TQA and NQ as TQA-matched and NQ-matched, respectively. The dataset sizes of all QA tasks are summarized in Appendix B.

4.2 Results

The results for closed-book QA tasks are summarized in Table 2. SKILL pre-trained models show improvements on FreebaseQA, WikiHop, and Wikidata-answerable versions of TriviaQA and NaturalQuestions, but no significant improvement on original TriviaQA and NaturalQuestions. As discussed in previous section, we believe this is due to the misalignment between the datasets and Wikidata.

Models pre-trained on Wikidata KG gives competitive results with ones on KELM sentences. It shows that the triple representation is as good as natural language representation, while being much easier to scale up for larger KG.

For T5.1.1-base and -large, additional pre-training on C4 boosts performance in comparison to the original baseline. For T5.1.1-XXL, this additional pre-training leads to a performance regress. In (Raffel et al., 2019), it is mentioned that training on C4 for multiple times may reduce the performance of a T5 model.

Impact of model size. As shown in Figure 1, SKILL pre-training introduces bigger improvements when applied on larger models. With more than 35M triples in Wikidata KG, it is harder for

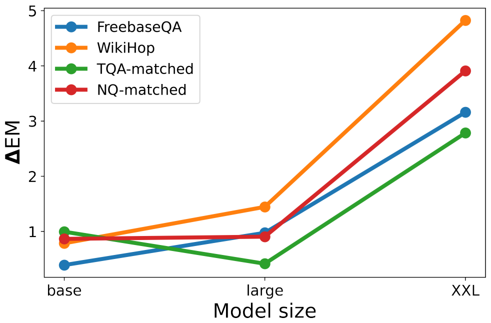

Figure 1: Performance improvements on closed-book QA tasks for different model sizes. The improvements are measured by the difference of exact match score (ΔEM) between knowledge-infused model trained with Wikidata triples and the baseline trained with C4 corpus.

Dataset	Split	Baseline	+ C4	+ KG
1-hop	dev	24.3	23.12	**71.52**
	test	24.5	23.53	**71.47**
2-hop	dev	32.05	32.23	**33.49**
	test	32.65	32.78	**33.57**
3-hop	dev	42.08	39.22	**43.79**
	test	42.31	39.66	**43.41**

Table 3: Exact match scores achieved by fine-tuning different T5.1.1-large checkpoints on MetaQA task.

smaller size models, e.g. T5.1.1.-base with 300M parameters, to memorize them efficiently. We view this as an encouraging result, suggesting that as model size grows, gains from SKILL pre-training may increase further.

Performance on a smaller KG. The WikiMovies KG (Miller et al., 2016) contains 134, 741 triples. T5.1.1-large should have enough parameters to memorize the KG. We train a T5.1.1-large model on the KG for 100K steps, ∼ 380 epochs, with the same hyperparameters as for Wikidata KG. We evaluate the checkpoints with MetaQA (Zhang et al., 2018) benchmark that was constructed over WikiMovies KG. The benchmark contains 3 different sub-tasks: 1-hop QA (e.g. "What films does Paresh Rawal appear in?"), 2-hop QA (e.g. "Who are the directors of the films written by Laura Kerr?"), 3-hop QA (e.g. "Who directed the movies written by the writer of Millennium Actress?").

The results in Table 3 demonstrate the effectiveness of SKILL pre-training, when it's possible to memorize the whole knowledge graph.

As 1-hop questions are supported by single triples in the WikiMovies KG, a 3× improvement on EM score is observed for the sub-task. In order to answer 2/3-hop questions it is not enough to memorize the triples, the model needs to be able to reason with them. This requires a better understanding of the graph structure. Training with single triples may not be enough, and the observed improvement is notably smaller. The performance could be further improved by representing more explicitly the graph structure in the training data, which we leave for future work.

5 Conclusion

We proposed a method to directly infuse knowledge from knowledge graphs into T5 models through pre-training. Empirical results show that T5 can learn directly from structured data and apply the learned knowledge to improve closed-book QA results. We also demonstrated that the models pre-trained on factual triples perform competitively with the ones on natural language sentences that contain the same knowledge. By enabling knowledge infusion directly from triples, this method can be very easily applied to industry-scale KGs.

6 Ethical and Broader Impact

In this work, we are introducing a new method to pre-train a well known natural language understanding model, T5, on the full corpora of public knowledge graphs. To the best of our knowledge, the method will not introduce extra bias to either the model or the dataset beyond the one potentially inherited from Wikidata (Vrandečić and Krötzsch, 2014) and WikiMovies (Miller et al., 2016) knowledge graphs. On the other hand, through knowledge fusion pre-training introduced in this work, a language model will be able to learn factual information to improve the quality of parameterized knowledge embedded in the model, which is demonstrated by improvements on various closed-book question-answering tasks. The proposed method and recipe will provide positive impact to the natural language processing community and help to improve the factualness in pre-trained large language model checkpoints.

Limitations. A factual triple is the basic ingredient of a knowledge graph. However, as a semantic network, the graph structure of a knowledge graph describes how the factual triples are connected. This information is not easy to directly represent by random set of triples. We leave the exploration of how to infuse the information implied by the graph structure for future work. We expect that this will further improve the results, especially for multi-hop question-answering tasks.

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.

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 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, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. In *Advances in Neural Information Processing Systems*, volume 33.

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.

Thibault Févry, Livio Baldini Soares, Nicholas FitzGerald, Eunsol Choi, and Tom Kwiatkowski. 2020. Entities as experts: Sparse memory access with entity supervision. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4937–4951, Online. Association for Computational Linguistics.

Kelvin Guu, Kenton Lee, Zora Tung, Panupong Pasupat, and Mingwei Chang. 2020. Retrieval augmented language model pre-training. In *International Conference on Machine Learning*, pages 3929–3938. PMLR.

Bin He, Di Zhou, Jinghui Xiao, Xin Jiang, Qun Liu, Nicholas Jing Yuan, and Tong Xu. 2020. BERT-MK: Integrating graph contextualized knowledge into pre-trained language models. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 2281–2290, Online. Association for Computational Linguistics.

Benjamin Heinzerling and Kentaro Inui. 2021. Language models as knowledge bases: On entity representations, storage capacity, and paraphrased queries. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 1772–1791, Online. Association for Computational Linguistics.

Gautier Izacard and Edouard Grave. 2021. Leveraging passage retrieval with generative models for open domain question answering. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 874–880, Online. Association for Computational Linguistics.

Shaoxiong Ji, Shirui Pan, Erik Cambria, Pekka Marttinen, and Philip S. Yu. 2020. A survey on knowledge graphs: Representation, acquisition and applications. *CoRR*, abs/2002.00388.

Kelvin Jiang, Dekun Wu, and Hui Jiang. 2019. FreebaseQA: A new factoid QA data set matching trivia-style question-answer pairs with Freebase. 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 318–323, Minneapolis, Minnesota. Association for Computational Linguistics.

Mandar Joshi, Eunsol Choi, Daniel Weld, and Luke Zettlemoyer. 2017. TriviaQA: A large scale distantly supervised challenge dataset for reading comprehension. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1601–1611, Vancouver, Canada. 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.

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.

George A. Miller. 1995. Wordnet: A lexical database for english. *Commun. ACM*, 38(11):39–41.

Barlas Oguz, Xilun Chen, Vladimir Karpukhin, Stan Peshterliev, Dmytro Okhonko, Michael Schlichtkrull, Sonal Gupta, Yashar Mehdad, and Scott Yih. 2020. UniK-QA: Unified representations of structured and unstructured knowledge for open-domain question answering. *arXiv preprint arXiv:2012.14610*.

Matthew E. Peters, Mark Neumann, Robert Logan, Roy Schwartz, Vidur Joshi, Sameer Singh, and Noah A. Smith. 2019. Knowledge enhanced contextual word representations. 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 43–54, Hong Kong, China. Association for Computational Linguistics.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2019. Exploring the limits of transfer learning with a unified text-to-text transformer. *CoRR*, abs/1910.10683.

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.

Sebastian Ruder, Matthew E. Peters, Swabha Swayamdipta, and Thomas Wolf. 2019. Transfer learning in natural language processing. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Tutorials*, pages 15–18, Minneapolis, Minnesota. Association for Computational Linguistics.

Maarten Sap, Ronan Le Bras, Emily Allaway, Chandra Bhagavatula, Nicholas Lourie, Hannah Rashkin, Brendan Roof, Noah A Smith, and Yejin Choi. 2019. Atomic: An atlas of machine commonsense for if-then reasoning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33.

Noam Shazeer and Mitchell Stern. 2018. Adafactor: Adaptive learning rates with sublinear memory cost. *CoRR*, abs/1804.04235.

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*.

Yu Sun, Shuohuan Wang, Shikun Feng, Siyu Ding, Chao Pang, Junyuan Shang, Jiaxiang Liu, Xuyi Chen, Yanbin Zhao, Yuxiang Lu, et al. 2021. Ernie 3.0: Large-scale knowledge enhanced pre-training for language understanding and generation. *arXiv preprint arXiv:2107.02137*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

Pat Verga, Haitian Sun, Livio Baldini Soares, and William Cohen. 2021. Adaptable and interpretable neural MemoryOver symbolic knowledge. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3678–3691, Online. Association for Computational Linguistics.

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

Johannes Welbl, Pontus Stenetorp, and Sebastian Riedel. 2018. Constructing datasets for multi-hop reading comprehension across documents. *Transactions of the Association for Computational Linguistics*, 6:287–302.

Donghan Yu, Chenguang Zhu, Yuwei Fang, Wenhao Yu, Shuohang Wang, Yichong Xu, Xiang Ren, Yiming Yang, and Michael Zeng. 2021. Kg-fid: Infusing knowledge graph in fusion-in-decoder for open-domain question answering. *arXiv preprint arXiv:2110.04330*.

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

A Matching of entities in KELM sentences

To find Wikidata KG entities in corresponding KELM sentences, we use Algorithm 1. Additional cycle on line 22 is needed because some entities have an information in brackets that should not be in a sentence, for example `John Doe (born 1990)`. This algorithm matched at least one entity to $15,383,248$ out of $15,628,486$ KELM sentences.

We don't try to match *relation* part of triples, because it could be represented in many different forms. For example, the triple (`Pulp Fiction`, `cast member`, `John Travolta`) could be represented as `"John Travolta was an actor in Pulp Fiction"`, `"John Travolta starred in Pulp Fiction"`, `"John Travolta played Vincent Vega in Pulp Fiction"`, etc., and there is no way to robustly align a relation to all possible surface forms.

Algorithm 1 KELM-Wikidata matching algorithm that finds spans in KELM sentences corresponding to Wikidata KG entities. $a \subset b$ means that a is a substring of b. $*$ represents any string.

```
$KELM_{matched} \leftarrow \emptyset$
for each $k \in KELM$ sentences do
    for each $t \in triples(k)$ do
        for each $e \in entities(t)$ do
            $e_p \leftarrow$ PREPROCESS($e$)
            $k_p \leftarrow$ PREPROCESS($k$)
            $spans \leftarrow$ MATCHENTITY($e_p$, $k_p$)
            $KELM_{matched}.insert([k, spans])$
        end for
    end for
end for

function MATCHENTITY($e$: entity, $k$: KELM sentence)
    $spans \leftarrow \emptyset$
    for each $s \subset k : date(e) = date(s)$ do
        $spans.insert(s)$
    end for
    for each $\exists s \subset k : e = s$ do
        $spans.insert(s)$
    end for
    if $spans = \emptyset$ then
        for each $\exists s \subset k : e = s+$" (*)" do
            $spans.insert(s)$
        end for
    end if
    return $spans$
end function

function PREPROCESS($str$: string)
    $str \leftarrow Lowercase(str)$
    $str \leftarrow RemovePunctuation(str)$
    return $str$
end function
```

B Dataset

Wikidata (Vrandečić and Krötzsch, 2014) was released under the Creative Commons CC0 License. KELM (Agarwal et al., 2021) was released under the Creative Commons CC BY-SA 2.0 License. NaturalQuestions (Kwiatkowski et al., 2019) and WikiHop (Welbl et al., 2018) were released under Creative Commons CC BY-SA 3.0 License. MetaQA (Zhang et al., 2018) was released under Creative Commons CC BY-ND 3.0 License. C4 (Raffel et al., 2019) and TriviaQA (Joshi et al., 2017) were released under Apache-2.0 License. WikiMovies (Miller et al., 2016) was released under MIT License. FreebaseQA (Jiang et al., 2019)[4] was released without a license.

	train	**dev**	**test**
FreebaseQA	20,358	3,994	3,996
WikiHop	39,364	4,374	5,129
TQA	78,785	8,837	11,313
TQA-matched	20,948	2,289	3,064
NQ	79,168	8,757	3,610
NQ-matched	10,487	1,160	462
MetaQA-1hop	96,106	9,992	9,947
MetaQA-2hop	118,980	14,872	14,872
MetaQA-3hop	114,196	14,274	14,274

Table 4: Dataset sizes for the closed-book QA tasks. TQA and NQ stands for TriviaQA and NaturalQuestions, respectively. *-matched are the selected dataset with the Wikidata KG answerable questions, and the KG alignment details can be found in Section 4.1.

[4] https://github.com/kelvin-jiang/FreebaseQA