Knowledge Graph Based Synthetic Corpus Generation for Knowledge-Enhanced Language Model Pre-training

Oshin Agarwal[*1] **Heming Ge**[2] **Siamak Shakeri**[2] **Rami Al-Rfou**[2]
[1] University of Pennsylvania [2]Google Research
oagarwal@seas.upenn.edu, {hemingge, siamaks, rmyeid}@google.com

Abstract

Prior work on Data-To-Text Generation, the task of converting knowledge graph (KG) triples into natural text, focused on domain-specific benchmark datasets. In this paper, however, we verbalize the entire English Wikidata KG, and discuss the unique challenges associated with a broad, open-domain, large-scale verbalization. We further show that verbalizing a comprehensive, encyclopedic KG like Wikidata can be used to integrate structured KGs and natural language corpora. In contrast to the many architectures that have been developed to integrate these two sources, our approach converts the KG into natural text, allowing it to be seamlessly integrated into existing language models. It carries the further advantages of improved factual accuracy and reduced toxicity in the resulting language model. We evaluate this approach by augmenting the retrieval corpus in a retrieval language model and showing significant improvements on the knowledge intensive tasks of open domain QA and the LAMA knowledge probe.

1 Introduction

Data-To-Text Generation (Kukich, 1983; Goldberg et al., 1994) involves converting knowledge graph (KG) triples of the form `(subject, relation, object)` into a natural language sentence(s). There are many standard datasets for this task such as WebNLG (Gardent et al., 2017) and many systems have been developed to improve performance on these datasets. However, to the best of our knowledge, no prior work has attempted to verbalize a full knowledge graph. Verbalizing a full KG has additional challenges over small benchmark datasets, such as entity and relation coverage and the lack of grouped sets of triples that can produce coherent sentences together. In this paper, we convert the *English* Wikidata KG (Vrandečić and Krötzsch, 2014) into natural language text (Figure 1). The

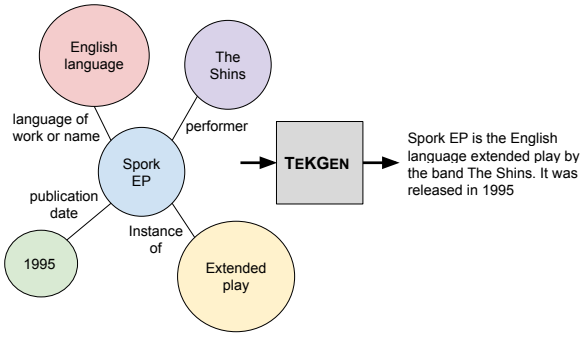

Figure 1: An example of generating text from KG. First, the entity subgraphs on the left are created and then converted to the sentence on the right.

generated corpus, which we call the KELM Corpus, consists of ∼18M sentences spanning ∼45M triples with ∼1500 distinct relations. For training the verbalization system, we also create an *English* Wikidata KG–Wikipedia Text aligned corpus consisting of a variety of entities such as dates and numerical quantities.

We evaluate the quality of the generated corpus through human evaluation of a random sample. We further showcase the utility of this corpus in language model pre-training. Text represents a limited coverage of the world knowledge. Therefore, we expect the language models to be restricted to facts that are expressed in natural language. Moreover, facts may not be expressed as explicitly in text as they are in KGs, and the variability in the quality of text can eventually cause biases in the resulting models (Bolukbasi et al., 2016; Sheng et al., 2019; Manzini et al., 2019). Building models that handle structured data and free form text seamlessly has been a long sought-after goal. However, their integration is challenging due to different structural formats. KG verbalization provides a simple way to integrate KGs with natural text. We illustrate this by augmenting the REALM (Guu et al., 2020) retrieval corpus with the KELM Corpus. We evaluate

[*]Work done during internship at Google

Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pages 3554–3565
June 6–11, 2021.

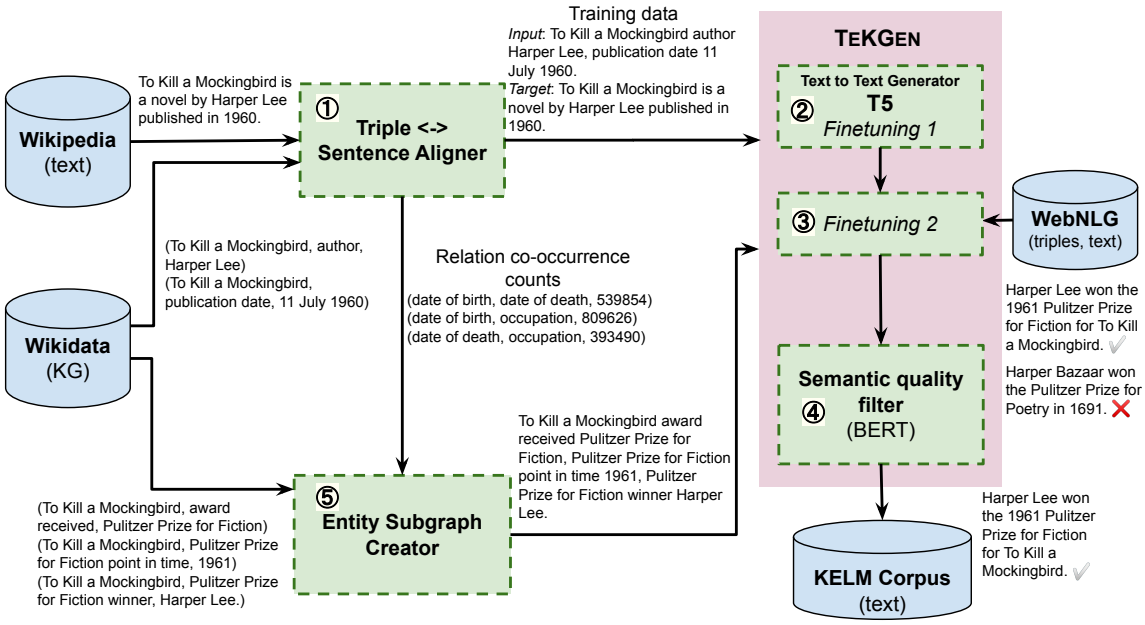

Figure 2: Pipelines for training the TEKGEN model and generating the KELM corpus. In Step ①, KG triples are aligned with Wikipedia text using distant supervision. In Steps ② & ③, T5 is fine-tuned sequentially first on this corpus, followed by a small number of steps on the WebNLG corpus, In Step ④, BERT is fine-tuned to generate a semantic quality score for generated sentences w.r.t. triples. Steps ②, ③ & ④ together form TEKGEN. To generate the KELM corpus, in Step ⑤, entity subgraphs are created using the relation pair alignment counts from the training corpus generated in step ①. The subgraph triples are then converted into natural text using TEKGEN.

the augmented system on the LAMA knowledge probe (Petroni et al., 2019) and open domain QA and show improvements on both. Through ablation experiments where we augment the retrieval corpus with the raw triples instead, we further confirm the effectiveness of verbalization. Our contributions are as follows -

- TEKGEN (**Te**xt from **KG Gen**erator): A data-to-text sequence-to-sequence model for verbalizing an entire KG
- TEKGEN training corpus: Text–KG aligned corpora with a wide variety of relations including dates and quantities
- KELM Corpus, [1] (Corpus for **K**nowledge-**E**nhanced **L**anguage **M**odel Pre-training): A large-scale synthetic corpus of Wikidata KG as natural text
- Data-to-text generation as a method to integrate KGs with textual pre-training corpora, showing improvements on open domain QA and LAMA probe with the augmented model

[1]Both the TEKGEN training corpus and the KELM corpus are available at https://github.com/google-research-datasets/KELM-corpus

2 TEKGEN

One of the challenges in converting an entire KG to text is the wide variety of entities and relations. Wikidata consists of ∼6M entities and ∼1500 relations. In comparison, the WebNLG dataset has ∼600 entities and ∼20 relations. In this section, we discuss the various components of TEKGEN, also illustrated in Figure 2 –

1. Create a large yet noisy training dataset using distant supervision.
2. Sequentially fine-tune T5, first on the dataset from step 1 for improved coverage, then on a small clean dataset for reduced hallucination.
3. Build a filter for the generated text based on its semantic quality w.r.t. the KG triples.

2.1 Training Dataset

We first create training data using distant supervision by aligning Wikidata triples to Wikipedia text (see Figure 3).

2.1.1 KG–Text Alignment

For each entity, we constrain the candidate sentences to the root section of its Wikipedia page

```
alignment_pairs ← {}
for all sentences t ∈ root section of Wiki page of entity s
do
    for all triples (s, r, o) ∈ KG do
        if t.contains(alias(o)) then
            if t.notcontains(alias(s)) then
                p ← t.first_pronoun
                t ← t.replace(p, name(s))
            end if
            alignment_pairs.add((t, (s, r, o)))
        end if
    end for
end for
```

Figure 3: KG–Text alignment algorithm.

because this section generally describes the relations of the subject entity with other entities. For each sentence in this section, we match all triples that have this entity as the subject. A triple is said to match if any alias of the object entity occurs in the sentence. We do not match relations to text as there are too many ways to express them. Constraining to the subject entity's page and root section generally ensures that the relation is expressed in the sentence if it mentions the object entity. Each triple can align to multiple sentences and each sentence can have multiple triples aligned to it. If any alias of the subject entity occurs in the given sentence, the sentence is selected as is, else the first animate third-person personal or possessive pronoun is replaced by the subject entity's canonical name. The pronoun replacement heuristic also works well because of this constraint. All triples aligned to a given sentences are combined together as a single example.

Alignment statistics are shown in Table 1 and some alignment examples are shown in Table 2. There are a total of ∼45M triples, ∼35% of which were aligned to sentences. This results in ∼8M examples, covering ∼42% of the relations.

Note that each sentence in the aligned corpus is matched to triples with a common subject entity. While this results in some noise, such errors should be small due to the constraint that the text is the root section of the subject entity page. This constraint allows us to maintain the same property of common subject entity as the entity subgraph used in inference (§3). It also simplified the alignment process, removing the need to match relations to text. In comparison, the T-REx (Elsahar et al., 2018) corpus does not have this noise due the use of typical NLP pipeline with coreference resolution and predicate linking. However, it suffers from errors due to entity linking and incorrect entailment, which are unlikely in our corpus due to this constraint.

Total KG triples	45,578,261
Triples aligned	16,090,457
Total sentences aligned	7,978,814
Total KG relations	1,522
Relations aligned	663

Table 1: KG–Text alignment statistics.

2.1.2 Types of Triples

We extract several types of triples, each of which have slightly different matching techniques. Other alignment corpora built using Wikipedia hyperlinks (Chen et al., 2020; Logan et al., 2019) would miss many of these triples with entities without Wikipedia pages such as quantities, dates and certain occupations, and hence relations such as date of birth, publication year and distance from Earth.

1. Object entity with a Wikipedia page: These are aligned by string matching all aliases. (e.g. Barack Obama)
2. Object entity without a Wikipedia page: These are also aligned by matching all aliases. (e.g. skateboarder, professional wrestler)
3. Object entity is a quantity: They have two components – Amount and Units. Units are also entities in the KG and have aliases. We concatenate the amount with each of the unit's aliases for matching (e.g. 16 km/hr, 16 km/h, 16 kilometres per hour). Certain quantities do not have units (e.g. When the relation is number of episodes).
4. Object entity is a date: Wikipedia uses only three date formats. [2] We first find all dates in a sentence using regular expressions and parse them into a structured format containing day of the month, month and year. If any of these components exist in both the dates to be matched, they should match. For example, if the triple date has all three components but the date extracted from a sentence has only the year, then only the year needs to match.
5. Relations with a subproperty: For instance, the relation *award received* has the subproperty *year* and the relation *spouse* may have the subproperty *place of marriage*.

[2] https://en.wikipedia.org/wiki/Wikipedia:Date_formattings

Input Triples	Target Sentence
Das Tagebuch der Anne Frank, (distributor, Universal Pictures), (country, Germany), (publication date, 03 March 2016)	The film was theatrically released in the Germany on March 3, 2016, by Universal Pictures International.
Neff Maiava, (date of birth, 01 May 1924), (date of death, 21 April 2018), (occupation, professional wrestler)	Maiava (May 1, 1924 April 21, 2018) was an American Samoan professional wrestler.
Barack Obama 2012 presidential campaign, (country, United States), (end time, 06 November 2012), (start time, 04 April 2011)	The 2012 reelection campaign of Barack Obama, the 44th President of the United States, was formally announced on April 4, 2011.
Blue whale (parent taxon, Balaenoptera)	The blue whale (Balaenoptera musculus) is a marine mammal belonging to the baleen whale suborder Mysticeti.

Table 2: Examples of alignment (training data).

We retain the main triple as such and reformat the subproperty as a triple of the form `(subject_entity, object_entity subproperty_name, subproperty_value)` e.g. (Barack, spouse, Michelle) has the subproperty (place of marriage, Trinity Church). These are converted to (Barack, spouse, Michelle) and (Barack, Michelle place of marriage, Trinity Church).

While the type of the triples is important in the alignment process, the verbalization model is agnostic to the type and treats all triples the same.

2.2 Model

We perform a two-step sequential finetuning of the pre-trained T5-large (Raffel et al., 2020) model for converting triples to text. Triples are concatenated as `subject relation_1 object_1, ....relation_n object_n` for input to T5. The model is first fine-tuned on the aligned corpus for 5000 steps to increase the coverage of entities and relations. However, this results in the generation of Wikipedia-like sentences and hallucination if a certain expected input triple is missing. For example, Wikipedia sentences generally mention *date of birth*, *date of death*, *occupation* together. If the occupation is missing in the input, the system hallucinates a random occupation. "Neff Maiava date of birth 01 May 1924, date of death, 21 April 2018." generates "Neff Maiava (1 May 1924 - 21 April 2018) was an Albanian actor."; hallucinating a profession. To overcome this, we further fine-tune the model on WebNLG 2017 data for 500 steps. While WebNLG has low coverage, the information in the input triples matches the target sentence exactly. WebNLG also has a different sentence structure than Wikipedia. This reduces conformity to Wikipedia sentence structure and hence reduces hallucination. We use a learning rate of 0.001, a batch size of 1048576 tokens and a maximum decoding length of 256.

Pearson correlation	0.73
Spearman correlation	0.66
Kendall's Tau	0.51

Table 3: Semantic Filtering Evaluation.

2.3 Quality Filtering

We perform a semantic quality based filtering of the sentences generated by the triple-to-text module. This is a separate post-processing module used during inference and is not jointly optimized with the text generation module. A semantic quality score is assigned to each generated sentence w.r.t. the input triples that indicates whether or not the generated text captures the full meaning of the triples and does not hallucinate extra information. The score is generated using a BERT base uncased model with input of the form `[CLS] concatenated-triples [SEP] reference-or-generated-sentence`. It is fine-tuned for 1000 steps on the WebNLG 2017 human assessment data. The data contains system predictions submitted to the shared task rated on a scale of 1-3 for semantics and fluency. We use the semantics score and scale it to 0-1. We also add gold references with a score of 1. This results in 2706 examples, 90% of which are used for finetuning and the remaining for evaluation. High correlations are obtained between the predicted scores and human scores on the evaluation split (Table 3).

3 KELM Corpus

In this section, we utilize the TEKGEN model and filtering mechanism to build a synthetic corpus that captures the KG in natural language format.

3.1 Entity Subgraph

Datasets such as WebNLG have instances with grouped triples that can be expressed as a fluent sentence. Such groups are not available for a large KG and using one triple at a time for inference would lead to hallucination as training uses multi-

```
all_triple_sets ← {}
rel_pairs ← {}
depth ← 5
for all r_i ∈ KG do
    P ← {(r_j, c_ij) ∀(r_i, r_j, c_ij) ∈ train_align_counts}
    rel_pairs(r_i) ← maxheap(P)
end for
for all entities s ∈ KG do
    R ← {(r, o) ∀(s, r, o) ∈ KG}
    while R ≠ ∅ do
        triple_set ← {}
        (r_1, o_1) ← random(R)
        triple_set.add(s, r_1, o_1)
        R.remove(s, r_1, o_1)
        KG.remove(s, r_1, o_1)
        for i = 2 to depth do
            r_i ← NONE
            M ← rel_pairs(r_{i-1})
            while M ≠ ∅ do
                (r_j, c_ij) ← M.next
                if r_j ∈ R then
                    r_i ← r_j
                    (r_i, o_i) ← R.get(r_i)
                    triple_set.add(s, r_i, o_i)
                    R.remove(s, r_i, o_i)
                    KG.remove(s, r_i, o_i)
                    break
                end if
            end while
        end for
        all_triple_sets.add(triple_set)
    end while
end for
```

Figure 4: Entity Subgraph Creation Algorithm using relation co-occurrence counts based on relation–sentence alignment in the training data. Each entity subgraph consists of a maximum of five triples, all with the same subject entity. The first triple is chosen at random. The second triple is chosen such that its relation has the highest co-occurrence count with the relation in the first triple and so on.

ple triples per example. Therefore, we develop a strategy to create entity subgraphs based on relation co-occurrence counts i.e. frequency of alignment of two relations to the same sentence in the training data. The algorithm is shown in Figure 4. It produces ∼18M entity subgraphs from ∼45M triples so the final corpus will have 18M generated sentences corresponding to each entity subgraph.

3.2 Generation

For each entity subgraph, we concatenate all its triples as before. We perform top 5 sampling with a temperature of 0.5. The bottom 1% of the generated sentences are filtered out based on the semantic score assigned using the model in §2.3.

Model	Finetuning data	Inference data	Semantics mean	Semantics stdev	Fluency mean	Fluency stdev
T5-only	WebNLG	Triple	4.12	1.02	4.16	1.02
T5-only	WebNLG	Subgraph	3.97	1.14	4.15	0.87
— TEKGEN —		Subgraph	4.36	0.87	4.60	0.58

Table 4: Human evaluation of the generated corpora, on a scale of 1-5, for semantics and fluency.

3.3 Human Evaluation

Generation quality of the KELM Corpus is evaluated using human ratings on a random sample of 200 entity subgraphs. Automatic metrics such as BLEU (Papineni et al., 2002) or BERTscore (Zhang et al., 2019) cannot be used due to the lack of gold references. Following prior work, the generated text is rated for two aspects–fluency and semantics, on a scale of 1-5, where 1 means not at all fluent/does not capture meaning at all and 5 means completely fluent/fully captures meaning with no hallucination. We have eight annotators total and each example is rated by two of them. All annotators are linguists, NLP researchers or NLP practitioners and volunteered for the evaluation. We do not use any crowd sourcing platform. For each instance, scores of the two annotators are averaged to get the final rating. The Pearson correlation between the two sets of ratings is 0.56 for semantics and 0.43 for fluency.

We compare TEKGEN to two baseline systems. For both baselines, we fine-tune a T5-large model only on WebNLG 2017 data but use different inference input. For one system, we use one triple at a time as input, resulting in 524 instances from the 200 entity subgraphs. For the second, we use the entity subgraphs as input, resulting in 200 instances. Scores are shown in Table 4. Entity subgraphs during inference do not improve the mean scores but reduce the standard deviation of the fluency. In comparison, TEKGEN with inference using entity subgraphs improve both the semantics and fluency of the generated text. Both the mean scores are higher and the standard deviations are lower. It paraphrases canonical names of relations in the KG to more natural expressions more often. Some examples of generation using the two systems are shown in Table 5. In the second example, the relation ‘inception’ is paraphrased to ‘started’ using WebNLG_finetuning+Triple_Inference and ‘founded’ using TEKGEN+Subgraph_Inference, the latter being more appropriate for organizations.

For completeness, we evaluate two more base-

Input Triples	WebNLG_Finetuning + Triples_Inference	TEKGEN + Subgraph_Inference
(Michelle Obama, height, +71 inch)	Michelle Obama's height is +71 inch.	Michelle Obama is 71 inches tall.
(10x10 Photobooks, instance of, Nonprofit organization), (10x10 Photobooks inception, 00 2012)	The 10x10 Photobooks are the result of a non-profit organization. 10x10 Photobooks was started in 00 2012.	10x10 Photobooks, founded in 2012 is a nonprofit organization.
(Edu (footballer, born 1949), member of sports team, Tigres UANL) (Edu (footballer, born 1949), Tigres UANL end time, 01 January 1983) (Edu (footballer, born 1949), Tigres UANL start time, 01 January 1978)	Edu was born in 1949 and is a member of Tigres UANL. Edu (footballer , born in 1949) Tigres UANL's end time was 01 January 1983. Edu (footballer , born 1949) was at Tigres UANL from 01 January 1978.	Edu, who was born in 1949, played for Tigres UANL between 1978 and 1983.
(To Kill a Mockingbird, award received, Pulitzer Prize for Fiction) (To Kill a Mockingbird Pulitzer Prize for Fiction point in time 00 1961) (To Kill a Mockingbird Pulitzer Prize for Fiction winner Harper Lee)	To Kill a Mockingbird won the Pulitzer Prize for Fiction. To Kill a Mockingbird was Pulitzer Prize for Fiction, awarded in 00 1961. Harper Lee was the winner of the Pulitzer Prize for Fiction for To Kill a Mockingbird.	Harper Lee won the 1961 Pulitzer Prize for Fiction for To Kill a Mockingbird.
(Caucasus Mountains, country, Georgia (country)) (Caucasus Mountains, instance of, Mountain range) (Caucasus Mountains, country, Russia) (Caucasus Mountains, highest point, Mount Elbrus) (Caucasus Mountains, country, Armenia) (Caucasus Mountains, topic's main category, Category:Caucasus Mountains)	The Caucasus Mountains are located in Georgia. The Caucasus Mountains is an example of a Mountain range. Caucasus Mountains is in Russia. The highest point in the Caucasus Mountain -s is Mount Elbrus. Caucasus Mountains is in the country of Armenia. The Caucasus Mountains is categorised as a Caucasus Mountains.	The Caucasus Mountains are a mountain range found in Georgia, Armenia and Russia. Mount Elbrus is the highest point in the Caucasus Mountains.

Table 5: Examples of text generated by the final model in comparison to the model trained only on WebNLG.

line systems in which T5-large model is finetuned only on the KG–Text aligned corpus but use the two different inference inputs–single triple and entity subgraphs. One annotator rated the same sample for semantics. The former had an average score of 2.34 and the latter 2.73. Since these scores were very low, we did not pursue the evaluation of these systems further. The use of just the aligned corpus which is noisy to some extent results in the worst performing system out of all the methods.

4 Knowledge Enhanced LMs

In this section, we showcase an application of the generated KELM Corpus as a way to integrate KGs into natural text corpora for pre-training language models (LMs), as shown in Figure 5. We choose REALM (Guu et al., 2020) as a representative of the recently introduced family of retrieval language models and therefore we expect our work to be equally applicable to other such language models. We show gains on LAMA knowledge probe and open domain QA with augmentation. We also perform experiments where we integrate raw Wikidata triples instead of KELM corpus to confirm the effectiveness of verbalization.

4.1 Retrieval Language Models

REALM is a retrieval-based language model and uses two corpora for pre-training–a retrieval corpus and a pre-training corpus. During pre-training, a sentence is selected at random from the pre-training corpus and a random word or salient span (dates and entities) is masked in this sentence. Then using a joint representation of the masked sentence and each of the documents in the retrieval corpus, the masked word is predicted. In the finetuning stage, the model is provided with a query/question as input in place of masked sentence from the pre-training corpora. It retrieves a small set of documents from the retrieval corpus based on the vector similarity of the query and document representation and then selects a span of text from the retrieved documents as the answer.

4.2 KELM Documents

We group sentences in the KELM corpus by subject entities to create 5722974 (5.7M) documents. We call these KELM documents. We then replace/augment the retrieval corpus in REALM with these synthetic documents. KELM Corpus has only ∼286M words (∼14%) in comparison to ∼2B words in English Wikipedia.

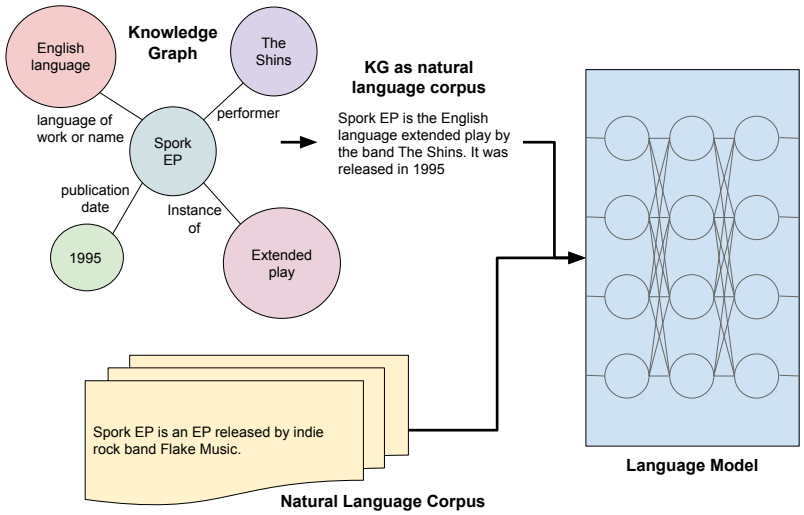

Figure 5: Knowledge Graph verbalization for integration with natural text corpora for language model pre-training.

4.3 Evaluation Datasets

We perform evaluation using two open domain question answering datasets and one knowledge probing dataset.

4.3.1 Question Answering

NaturalQuestions (NQ) (Kwiatkowski et al., 2019): Queries to Google and their answers.

WebQuestions (WQ) (Berant et al., 2010): question-answers from the Google Suggest API.

We keep the same settings as REALM for both NQ and WQ i.e. we work on the open domain setting for both datasets where *no* passage is provided as context for each question. Finetuning is performed on respective training splits.

4.3.2 Knowledge Probe

LAMA (Petroni et al., 2019): Fill-in-the-Blank style factual queries with single token answers from four sources: Google-RE,[3] T-REx (Elsahar et al., 2018), SQuAD (Rajpurkar et al., 2016) and ConceptNet (Speer and Havasi, 2012). Google-RE also consists of aliases of each answer.

REALM did not include LAMA as one of its evaluation datasets. So we first evaluate REALM on LAMA using the original retrieval corpus and then using the KELM Corpus. No finetuning is performed and the masked word predictions from the pre-trained models are used as answers.

[3] https://code.google.com/archive/p/relation-extraction-corpus/

4.4 Results

We evaluate REALM on WQ, NQ and LAMA under three settings by modifying the retrieval corpus.

1. ORIGINAL: Wikipedia text
2. REPLACED: only KELM Corpus
3. AUGMENTED: Wikipedia text + KELM Corpus

The REPLACED and AUGMENTED models are evaluated using both the raw Wikidata triples and the generated sentences. Wikidata triples are grouped by subject entity to form Triple Documents and KELM Corpus sentences are also grouped by subject entity to form KELM Corpus Documents (§4.2). The model is pre-trained for 200k steps with the CC-News pre-training corpus in all cases with default hyperparameters.

ORIGINAL For NQ and WQ, we fine-tuned the pre-trained REALM on the respective training splits. While we were able to reproduce the accuracy on WQ, the accuracy on NQ was ∼1.5% absolute less than the reported accuracy (row 1&2 in Table 7). For LAMA probe, we first evaluated the pre-trained REALM, reporting the results on the different sub-corpora in Table 6 (row *Wikipedia* under REALM). Even the ORIGINAL REALM model shows substantial improvement over prior models. The ability of REALM to access the corpus documents during inference not only make it interpretable but also better on the knowledge intensive tasks. It obtains an accuracy of 67.36% on Google-RE, 68.18% on T-REx and 27.96% on

	Google-RE				TREx				Squad	Concept Net
	DOB	POB	POD	Total	1-1	N-1	N-M	Total		
Elmo 5.5B (Peters et al., 2018)	0.10	7.50	1.30	3.00	13.10	6.50	7.40	7.10	4.30	6.20
Tranformer-XL (Dai et al., 2019)	0.90	1.10	2.70	1.60	36.50	18.00	16.50	18.30	3.90	5.70
BERT-large (Devlin et al., 2019)	1.40	16.10	14.00	10.50	**74.50**	34.20	24.30	32.30	17.40	**19.20**
REALM										
ORIGINAL										
Wikipedia	49.06	79.56	64.13	67.36	55.81	69.54	66.98	68.18	27.96	4.78
REPLACED										
Triple Documents	69.46	61.64	53.01	63.03	49.30	62.34	53.12	58.43	18.09	4.27
KELM Documents	68.91	61.37	53.79	62.81	49.41	61.60	52.50	57.76	19.07	4.26
AUGMENTED										
Wikipedia + Triple Documents	71.60	80.92	69.89	76.32	57.20	69.96	67.86	68.80	29.93	4.81
Wikipedia + KELM Documents	**76.75**	**83.92**	**74.86**	**80.30**	57.84	**70.33**	**68.09**	**69.13**	**31.57**	5.25

Table 6: Accuracy on LAMA probe. Pretaining corpus is CCnews and the retrieval corpus changed for REALM.

REALM Retrieval Corpus	**NQ**	**WQ**
ORIGINAL		
Wikipedia (reported)	40.40	40.70
Wikipedia (rerun)	38.84	40.80
REPLACED		
Triple Documents	21.14	42.54
KELM Documents	22.58	41.19
AUGMENTED		
Wikipedia + Triple Documents	40.28	42.91
Wikipedia + KELM Documents	**41.47**	**43.90**

Table 7: Exact Match (EM) accuracy of REALM on NQ and WQ. Pretraining corpus used is CC-News.

SQuAD. In comparison, the reported accuracy for BERT (Devlin et al., 2019) is 10.50% on Google-RE, 32.30% on T-REx and 17.40% on SQuAD. BERT performs better on 1-1 T-REx relations with 74.50% accuracy as compared to REALM with 55.81% accuracy. However, this group consists of only two relations; *capital* and *capital of*. BERT also has better performance than REALM on the ConceptNet subcorpus. On inspection of some of the queries in ConceptNet, we found the questions to be vague and possibly hard for even humans. For example, *Raven can* ___ and *Time is* ___.

REPLACED The REPLACED model which uses only KELM Corpus Documents, performs better than the ORIGINAL model on WQ but the accuracy is much lower on NQ (rows 3&4 in Table 7). This can be attributed to the nature of the datasets–WQ is a KG-based dataset whereas NQ consists of real queries issued to Google. On LAMA (rows 2&3 under REALM in Table 6), the performance is lower than the ORIGINAL model but much higher than BERT. Both Triple Documents and KELM Corpus Documents have similar performance. When using just the KG, the format doesn't matter. However, a system trained on raw triples may not generalize for tasks where sentence structure is important.

AUGMENTED We observe improvements on all the datasets (last two rows of Tables 6&7) with the AUGMENTED model which uses both the Wikipedia text and the KELM Documents. There is an absolute gain of 2.63% and 3.10% on NQ and WQ respectively over the ORIGINAL model. Similarly, there is an absolute gain of 12.94%, 0.95%, 3.61% and 0.47% on Google-RE, T-REx, SQuAD and ConceptNet in LAMA respectively. Unlike the REPLACED model, the improvement is higher when the generated sentences in KELM Corpus are added instead of the raw Wikidata triples, confirming the effectiveness of verbalization of KG into natural language sentences. Wikipedia is the dominant corpus with 2B words whereas KELM corpus sentences are succinct with a total of 286M words (§4.2) so it is likely the learned representations favour the Wikipedia format which is natural language sentences.

We expect augmenting other retrieval-based models such as DPR (Karpukhin et al., 2020) and RAG (Lewis et al., 2020) with the KELM corpus should also improve their performance, given that their enhancements are orthogonal to our contribution. Moreover, we note that our augmented corpus represents a scalable strategy for future QA systems; by adding only 14% more tokens to the original REALM model we outperform huge and computationally expensive models such as (Roberts et al., 2020) (11B parameters) on NQ (35.20 → 41.47) and WQ (42.80 → 43.90). Wikipedia is the dominant corpus with 2B words whereas KELM corpus sentences are succinct with a total of 286M words (§4.2) so it is likely the learned representations favour the Wikipedia format which is natural language sentences.

We inspected the errors of the AUGMENTED model with KELM Documents on LAMA. Apart

from real errors where the prediction is actually incorrect, there were some false errors that can be broadly classified into three categories–

1. Ambiguous Query: e.g. In "X was born in ____", the answer could be the year or the place of birth but only one of them is acceptable depending on the subcorpus.
2. Incomplete Answer Set: e.g. In "Konstantin Mereschkowski had a career as ____", the gold target is *biologist* and the prediction is *botanist* but both should be correct.
3. Answer granularity: The prediction is correct but more specific. e.g. In "On the CPI scale, Kenya ranks ____", the gold answer is *low* but the prediction is *101*, which is in fact correct.

5 Related Work

Data-to-Text Generation Data-to-Text Generation has several benchmark datasets with slightly different objectives–WebNLG (Gardent et al., 2017) to convert a group of triples to text, E2ENLG (Dušek et al., 2018) to convert database key-value pairs or pictures to text, WikiBio (Lebret et al., 2016) for biography generation from text, Wiseman et al. (2017) for text describing score statistics tables of basketball games, both ToTTo (Parikh et al., 2020) and DART (Radev et al., 2020) to generate text given a table and relevant highlighted cells. Many systems (van der Lee et al., 2018; Castro Ferreira et al., 2019; Shimorina and Gardent, 2018) have been developed and evaluated on these datasets, such as graph transformers over structured data (Koncel-Kedziorski et al., 2019), latent templates for interpretability (Wiseman et al., 2018) and text-to-text generation with T5 (Kale, 2020).

KG–Text alignment T-REx (Elsahar et al., 2018) is a widely used Text–KG aligned corpus, built using systems such as coreference resolution and predicate linkers (details in §2.1.1). Logan et al. (2019) and Chen et al. (2020) also created an aligned corpus using Wikipedia hyperlinks and coreference resolution. (details on comparison in §2.1.2). In contrast, we use alias-based heuristics coupled with source text selection constraints to generate a corpus of 16M triples aligned with 8M sentences. Lastly, open information extraction i.e. automatic KG construction from text (Etzioni et al., 2008; Angeli et al., 2015; Clancy et al., 2019) inherently create such a corpus but these works generally do not release the extracted KG triples.

Incorporating KGs Most prior works on incorporating KG with text often learn KG entity representations and add them to the mention spans linked to the entity (Peters et al., 2019; Yu et al., 2020; Févry et al., 2020) or create subgraphs relevant to the query that are expanded with text in the embedding space (Logan et al., 2019; Sun et al., 2019; Xiong et al., 2019). Some others incorporate additional modules. Verga et al. (2020) extend Févry et al. (2020) by adding a triple memory with (subject, relation) encoding as the key and the object encoding as the value. Das et al. (2017) use universal schema (Riedel et al., 2013) that embeds text and KGs in a shared space for their integration. K M et al. (2018) learn a single representation for all the triples mentioned in a sentences during pre-training and update it further in task-specific finetuning. In contrast, we convert the KG into text and use it to augment the pre-training data.

6 Future Work

The KELM corpus sentences covers all facts in the KG but the generated sentences are limited to a given entity and its direct relations to other entities. For example, given the triples (X, child, Y) and (Y, child, Z), it does not the contain "Z is a grandchild of X". More complex sentences could be generated by incorporating multi-hop relations in the KG. Recent work has also shown promising results on generating multilingual text from English triples (Castro Ferreira et al., 2020; Agarwal et al., 2020). Our proposed approach can be applied to generate a multilingual corpus of facts in various languages using English Wikidata.

7 Conclusion

In this paper, we converted an entire KG (Wikidata) to natural text (KELM Corpus), tackling various challenges over verbalizing domain-specific benchmark datasets. We further showcase that KG verbalization can be used to integrate KGs and natural text corpora by including the verbalized KG as additional pre-training data. We augment a retrieval-based language model with the generated synthetic KELM corpus as a retrieval corpus. We evaluated the augmented model on open domain QA and a knowledge probe, showing improvements on both. The KELM Corpus is publicly available at https://github.com/google-research-datasets/KELM-corpus.

Acknowledgments

We thank William Woods, Jonni Kanerva, Tania Rojas-Esponda, Jianmo Ni, Aaron Cohen and Itai Rolnick for rating the synthetic corpus sample for human evaluation. We also thank Kelvin Guu for his valuable feedback on the paper.

References

Oshin Agarwal, Mihir Kale, Heming Ge, Siamak Shakeri, and Rami Al-Rfou. 2020. Machine translation aided bilingual data-to-text generation and semantic parsing. In *Proceedings of the 3rd International Workshop on Natural Language Generation from the Semantic Web (WebNLG+)*, pages 125–130, Dublin, Ireland (Virtual). Association for Computational Linguistics.

Gabor Angeli, Melvin Jose Johnson Premkumar, and Christopher D. Manning. 2015. Leveraging linguistic structure for open domain information extraction. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 344–354, Beijing, China. Association for Computational Linguistics.

Jonathan Berant, Ido Dagan, and Jacob Goldberger. 2010. Global learning of focused entailment graphs. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 1220–1229, Uppsala, Sweden. Association for Computational Linguistics.

Tolga Bolukbasi, Kai-Wei Chang, James Y Zou, Venkatesh Saligrama, and Adam T Kalai. 2016. Man is to computer programmer as woman is to homemaker? debiasing word embeddings. In *Advances in neural information processing systems*, pages 4349–4357.

Thiago Castro Ferreira, Claire Gardent, Nikolai Ilinykh, Chris van der Lee, Simon Mille, Diego Moussallem, and Anastasia Shimorina. 2020. The 2020 bilingual, bi-directional WebNLG+ shared task: Overview and evaluation results (WebNLG+ 2020). In *Proceedings of the 3rd International Workshop on Natural Language Generation from the Semantic Web (WebNLG+)*, pages 55–76, Dublin, Ireland (Virtual). Association for Computational Linguistics.

Thiago Castro Ferreira, Chris van der Lee, Emiel van Miltenburg, and Emiel Krahmer. 2019. Neural data-to-text generation: A comparison between pipeline and end-to-end architectures. 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 552–562, Hong Kong, China. Association for Computational Linguistics.

Wenhu Chen, Yu Su, Xifeng Yan, and William Yang Wang. 2020. Kgpt: Knowledge-grounded pretraining for data-to-text generation. *arXiv preprint arXiv:2010.02307*.

Ryan Clancy, Ihab F. Ilyas, and Jimmy Lin. 2019. Scalable knowledge graph construction from text collections. In *Proceedings of the Second Workshop on Fact Extraction and VERification (FEVER)*, pages 39–46, Hong Kong, China. Association for Computational Linguistics.

Zihang Dai, Zhilin Yang, Yiming Yang, Jaime Carbonell, Quoc V Le, and Ruslan Salakhutdinov. 2019. Transformer-xl: Attentive language models beyond a fixed-length context. *arXiv preprint arXiv:1901.02860*.

Rajarshi Das, Manzil Zaheer, Siva Reddy, and Andrew McCallum. 2017. Question answering on knowledge bases and text using universal schema and memory networks. *arXiv preprint arXiv:1704.08384*.

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.

Ondřej Dušek, Jekaterina Novikova, and Verena Rieser. 2018. Findings of the e2e nlg challenge. *arXiv preprint arXiv:1810.01170*.

Hady Elsahar, Pavlos Vougiouklis, Arslen Remaci, Christophe Gravier, Jonathon Hare, Frederique Laforest, and Elena Simperl. 2018. T-REx: A large scale alignment of natural language with knowledge base triples. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan. European Language Resources Association (ELRA).

Oren Etzioni, Michele Banko, Stephen Soderland, and Daniel S Weld. 2008. Open information extraction from the web. *Communications of the ACM*, 51(12):68–74.

Thibault Févry, Livio Baldini Soares, Nicholas FitzGerald, Eunsol Choi, and Tom Kwiatkowski. 2020. Entities as experts: Sparse memory access with entity supervision. *arXiv preprint arXiv:2004.07202*.

Claire Gardent, Anastasia Shimorina, Shashi Narayan, and Laura Perez-Beltrachini. 2017. The WebNLG challenge: Generating text from RDF data. In *Proceedings of the 10th International Conference on Natural Language Generation*, pages 124–133, Santiago de Compostela, Spain. Association for Computational Linguistics.

E. Goldberg, N. Driedger, and R. I. Kittredge. 1994. Using natural-language processing to produce weather forecasts. *IEEE Expert*, 9(2):45–53.

Kelvin Guu, Kenton Lee, Zora Tung, Panupong Pasupat, and Ming-Wei Chang. 2020. Realm: Retrieval-augmented language model pre-training. *arXiv preprint arXiv:2002.08909*.

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.

Mihir Kale. 2020. Text-to-text pre-training for data-to-text tasks. *arXiv preprint arXiv:2005.10433*.

Vladimir Karpukhin, Barlas Oğuz, Sewon Min, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. 2020. Dense passage retrieval for open-domain question answering. *arXiv preprint arXiv:2004.04906*.

Rik Koncel-Kedziorski, Dhanush Bekal, Yi Luan, Mirella Lapata, and Hannaneh Hajishirzi. 2019. Text Generation from Knowledge Graphs with Graph Transformers. 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 2284–2293, Minneapolis, Minnesota. Association for Computational Linguistics.

Karen Kukich. 1983. Design of a knowledge-based report generator. In *21st Annual Meeting of the Association for Computational Linguistics*, pages 145–150, Cambridge, Massachusetts, USA. Association for Computational Linguistics.

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

Rémi Lebret, David Grangier, and Michael Auli. 2016. Neural text generation from structured data with application to the biography domain. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1203–1213, Austin, Texas. Association for Computational Linguistics.

Patrick Lewis, Ethan Perez, Aleksandara Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, et al. 2020. Retrieval-augmented generation for knowledge-intensive nlp tasks. *arXiv preprint arXiv:2005.11401*.

Robert Logan, Nelson F. Liu, Matthew E. Peters, Matt Gardner, and Sameer Singh. 2019. Barack's wife hillary: Using knowledge graphs for fact-aware language modeling. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5962–5971, Florence, Italy. Association for Computational Linguistics.

Thomas Manzini, Lim Yao Chong, Alan W Black, and Yulia Tsvetkov. 2019. Black is to criminal as caucasian is to police: Detecting and removing multiclass bias in word embeddings. 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 615–621, Minneapolis, Minnesota. Association for Computational Linguistics.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.

Ankur P Parikh, Xuezhi Wang, Sebastian Gehrmann, Manaal Faruqui, Bhuwan Dhingra, Diyi Yang, and Dipanjan Das. 2020. Totto: A controlled table-to-text generation dataset. *arXiv preprint arXiv:2004.14373*.

Matthew Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. 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 2227–2237, New Orleans, Louisiana. Association for Computational Linguistics.

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.

Fabio Petroni, Tim Rocktäschel, Sebastian Riedel, Patrick Lewis, Anton Bakhtin, Yuxiang Wu, and Alexander Miller. 2019. Language models as knowledge bases? In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 2463–2473, Hong Kong, China. Association for Computational Linguistics.

Dragomir Radev, Rui Zhang, Amrit Rau, Abhinand Sivaprasad, Chiachun Hsieh, Nazneen Fatema Rajani, Xiangru Tang, Aadit Vyas, Neha Verma, Pranav Krishna, et al. 2020. Dart: Open-domain structured data record to text generation. *arXiv preprint arXiv:2007.02871*.

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. *Journal of Machine Learning Research*, 21(140):1–67.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. SQuAD: 100,000+ questions for machine comprehension of text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2383–2392, Austin, Texas. Association for Computational Linguistics.

Sebastian Riedel, Limin Yao, Andrew McCallum, and Benjamin M. Marlin. 2013. Relation extraction with matrix factorization and universal schemas. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 74–84, Atlanta, Georgia. Association for Computational Linguistics.

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.

Emily Sheng, Kai-Wei Chang, Premkumar Natarajan, and Nanyun Peng. 2019. The woman worked as a babysitter: On biases in language generation. 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 3407–3412, Hong Kong, China. Association for Computational Linguistics.

Anastasia Shimorina and Claire Gardent. 2018. Handling rare items in data-to-text generation. In *Proceedings of the 11th International Conference on Natural Language Generation*, pages 360–370, Tilburg University, The Netherlands. Association for Computational Linguistics.

Robyn Speer and Catherine Havasi. 2012. Representing general relational knowledge in ConceptNet 5. In *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC'12)*, pages 3679–3686, Istanbul, Turkey. European Language Resources Association (ELRA).

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.

Chris van der Lee, Emiel Krahmer, and Sander Wubben. 2018. Automated learning of templates for data-to-text generation: comparing rule-based, statistical and neural methods. In *Proceedings of the 11th International Conference on Natural Language Generation*, pages 35–45, Tilburg University, The Netherlands. Association for Computational Linguistics.

Pat Verga, Haitian Sun, Livio Baldini Soares, and William W Cohen. 2020. Facts as experts: Adaptable and interpretable neural memory over symbolic knowledge. *arXiv preprint arXiv:2007.00849*.

Denny Vrandečić and Markus Krötzsch. 2014. Wikidata: A free collaborative knowledge base. *Communications of the ACM*, 57:78–85.

Sam Wiseman, Stuart Shieber, and Alexander Rush. 2017. Challenges in data-to-document generation. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2253–2263, Copenhagen, Denmark. Association for Computational Linguistics.

Sam Wiseman, Stuart M Shieber, and Alexander M Rush. 2018. Learning neural templates for text generation. *arXiv preprint arXiv:1808.10122*.

Wenhan Xiong, Mo Yu, Shiyu Chang, Xiaoxiao Guo, and William Yang Wang. 2019. Improving question answering over incomplete KBs with knowledge-aware reader. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4258–4264, Florence, Italy. Association for Computational Linguistics.

Donghan Yu, Chenguang Zhu, Yiming Yang, and Michael Zeng. 2020. Jaket: Joint pre-training of knowledge graph and language understanding. *arXiv preprint arXiv:2010.00796*.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q Weinberger, and Yoav Artzi. 2019. Bertscore: Evaluating text generation with bert. *arXiv preprint arXiv:1904.09675*.