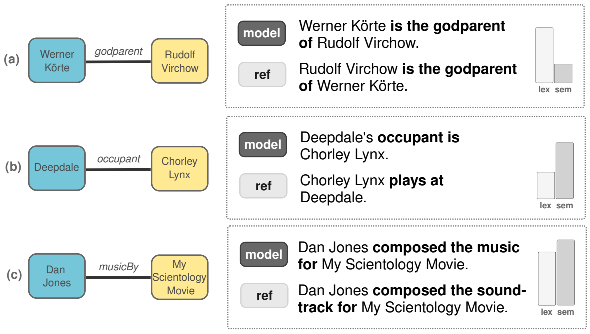
Mind the Labels: Describing Relations in Knowledge Graphs With Pretrained Models

Zdeněk Kasner,[1] **Ioannis Konstas**[2] and **Ondřej Dušek**[1]
[1]Charles University, Faculty of Mathematics and Physics, Prague, Czechia
[2]The Interaction Lab, MACS, Heriot-Watt University, Edinburgh, UK
{kasner,odusek}@ufal.mff.cuni.cz, i.konstas@hw.ac.uk

Abstract

Pretrained language models (PLMs) for data-to-text (D2T) generation can use *human-readable data labels* such as column headings, keys, or relation names to generalize to out-of-domain examples. However, the models are well-known in producing semantically inaccurate outputs if these labels are ambiguous or incomplete, which is often the case in D2T datasets. In this paper, we expose this issue on the task of descibing a relation between two entities. For our experiments, we collect a novel dataset for verbalizing a diverse set of 1,522 unique relations from three large-scale knowledge graphs (Wikidata, DBPedia, YAGO). We find that although PLMs for D2T generation expectedly fail on unclear cases, models trained with a large variety of relation labels are surprisingly robust in verbalizing novel, unseen relations. We argue that using data with a diverse set of clear and meaningful labels is key to training D2T generation systems capable of generalizing to novel domains.[1]



model: Werner Körte **is the godparent of** Rudolf Virchow.

ref: Rudolf Virchow **is the godparent of** Werner Körte.

model: Deepdale's **occupant is** Chorley Lynx.

ref: Chorley Lynx **plays at** Deepdale.

model: Dan Jones **composed the music for** My Scientology Movie.

ref: Dan Jones **composed the soundtrack for** My Scientology Movie.

Figure 1: The data-to-text generation models use the relation labels (such as *godparent*, *occupant*, and *musicBy*) to describe the relation between the entities. However, unclear labels can lead to various lexical or semantic incoherencies in the output descriptions, such as swapping the relation direction (a) or using too literal expressions (b).

1 Introduction

D2T generation systems need to accurately capture the semantics of relations between values in the data. However, the data labels such as relation names (Färber et al., 2018; Haller et al., 2022), table headings (Parikh et al., 2020), or meaning representation keys (Dušek et al., 2020) may provide only superficial or—if the labels are abbreviations, such as in the Rotowire dataset (Wiseman et al., 2017)—no usable hints about the data semantics. Learning how to properly describe the data is thus a challenge for D2T systems, typically requiring in-domain training data of sufficient quality and quantity (Dušek et al., 2019).

PLMs such as BART (Lewis et al., 2020) or T5 (Raffel et al., 2020) can quickly adapt to new domains and exhibit robustness to out-of-domain inputs. However, the PLMs for D2T generation are still limited by the expressivity of the data labels. Consider Figure 1 (a): the model can use its representation of *"godparent"* to understand there is a *"is-a-godparent-of"* relation between the entities, but it has to infer (or guess) who is the godparent of whom. Even in the less ambiguous cases (b) and (c), the model still has to correctly capture the intended semantics of the relation (e.g. *"occupant"* meaning *"home team"*).

In this paper, we investigate to what extent PLMs are able to use arbitrary labels describing relations between entities. A suitable testing ground is the task of describing (i.e., *verbalizing*) individual triples in a knowledge graph (KGs), which can be considered a trivial case of graph-to-text (G2T) generation (Ribeiro et al., 2020a; Koncel-Kedziorski et al., 2019). In this task, there is a wide range of lexical choices for the *relation label* (see Table 1), while the *entities* can be copied verbatim or with only minor morphological changes.

Current human-annotated datasets for D2T generation contain only a small number of relations

[1] We release the code and data for our experiments in this anonymized repository: https://anonymous.4open.science/r/rel2text/.

relation	possible verbalization
is part of	X is part of Y.
duration	X lasted for Y.
platform	X is available on Y. X runs on Y.
country	X was born in Y. X is located in Y.
parent	X is the parent of Y. Y is the parent of X.
ChEMBL	X has an id Y in the ChEMBL database.

Table 1: Examples of relation labels and their possible verbalizations, with placeholders for head (X) and tail (Y) entities. Relations can be copied verbatim (*is part of*), have a unique verbalization (*duration*), or multiple equivalent lexical choices (*platform*). There is also ambiguity stemming from the semantics of the entities (*country*) or the relation itself (*parent*, *ChEMBL*).

and rarely contain any unseen relations in the test set (Mille et al., 2021). We collect a novel dataset REL2TEXT (Re-writing edge labels to Text)[2], acting as a test bench for our experiments. It contains 4,097 single triples from three large-scale KGs (Wikidata, DBPedia, and YAGO) and their crowdsourced verbalizations, covering 1,522 unique relations (§3). Each relation is equipped with a label, a textual description, and up to five triples in which the relation occurs in the KG.

Using the REL2TEXT dataset, we evalute the ability of PLMs to verbalize relations which were not present in the training set. We consider both models finetuned on other relations in our dataset and models finetuned on datasets from a related domain. We also experiment with scenarios involving few-shot finetuning, training on masked labels, and extending the labels with descriptions (§4, 5).

We find that the PLMs are quite robust in verbalizing a diverse set of relations based on their label (achieving ~90% of overall entailment probability). We show that semantically unfaithful model outputs are often caused by incomplete, ambiguous, or noisy input data. Somewhat suprisingly, we also show that longer relation descriptions do not provide substantial improvements over using short labels. However, even for data using short relation labels, the model trained on verbalizing relations can achieve results comparable to verbalizing relations using manual templates in two downstream tasks (§6).

The contributions of our work are as follows:

- We examine the ability of PLMs to describe graph relations, showing that *clear and meaningful labels* are the basis for successful generalization to unseen relations.
- We present REL2TEXT—a human-annotated dataset with 4,097 examples verbalizing 1,522 relations from three large-scale open KGs.
- We show that a model trained on REL2TEXT can serve as a drop-in replacement for manual templates, preserving or improving performance on downstream tasks.

2 Related Work

Earlier works in natural language generation from KGs exploited domain-specific ontologies for rule-based systems (Cimiano et al., 2013; Bouayad-Agha et al., 2012; Sun and Mellish, 2007, 2006). With the advance of PLMs, structure-aware modeling and task-specific pretraining has lead to remarkable progress on **D2T benchmarks** such as WebNLG (Gardent et al., 2017b; Ferreira et al., 2020), AGENDA (Koncel-Kedziorski et al., 2019), or E2E (Dušek et al., 2020), indicated via both automatic and human evaluation metrics (Ke et al., 2021; Guo et al., 2020; Ribeiro et al., 2020b; Harkous et al., 2020).

Recently, Agarwal et al. (2021) used a multi-step approach with semantic filtering and distant supervision for **verbalizing the English Wikidata**, covering the wide range of relations present in the KG. The authors use the approach to generate the KeLM corpus – an automatically cleaned corpus with *synthetic* (model-generated) verbalizations of Wikidata triplesets. We use the KeLM corpus to investigate how the models trained on large-scale synthetic data differ from models trained on a small-scale human-annotated dataset (cf. §4).

Other works have tried **incorporating descriptions of data labels** in the model inputs. In one of the experiments, Wang et al. (2021) use descriptions of relations from Wikidata instead of their labels for relation embeddings, concluding that it results in worse performance on downstream tasks. Conversely, Kale and Rastogi (2020) and Lee et al. (2021) improve the performance of their systems by including schema descriptions on the input for the dialogue state tracking and dialogue response generation systems.

There has also been a research interest in **verbalizing single triples** as a stand-alone preprocessing step for natural language processing tasks. The

[2] Or simply “Relations-to-Text”.

step has been shown to improve the generalization ability of downstream models for data-to-text generation (Laha et al., 2019; Kasner and Dušek, 2020, 2022) and response generation in dialogue systems (Kale and Rastogi, 2020). This step can also serve for making the input similar to the format used during pretraining, e.g. for natural language inference (NLI) models (Gupta et al., 2020; Neeraja et al., 2021; Dušek and Kasner, 2020). All of the above works transform triples to text using either delexicalized human references or hand-crafted templates, ranging from simple look-up tables to rule-based systems.

In a work concurrent to ours, Keymanesh et al. (2022) investigate the aspects of **generalization performance of PLMs** on the DART dataset[3] (Nan et al., 2021). They compare prompt-based and finetuning-based approaches to D2T generation, focusing on the ability of models to perform on difficult examples. In contrast, we focus on finetuned encoder-decoder models, which were shown in Keymanesh et al. (2022) to be more efficient for D2T generation, and we evaluate the models on clean and manually curated data.

3 Data

For our experiments, we need data with diverse labels and their human verbalizations. In this section, we describe how we gather RDF[4] triples from large-scale KGs (§3.1) and collect their verbalization through crowdsourcing (§3.2, 3.3).

3.1 Input Data

An RDF triple is a tuple $t = (e_h, r, e_t)$, where r denotes the relation between the head entity e_h and the tail entity e_t. We retrieve triples from three open large-scale KGs encoding factual knowledge:

- **Wikidata** (Vrandecic and Krötzsch, 2014) is a large-scale Wikipedia-based KG created using collaborative editing. With approximately 10,000 human-created relations equipped with descriptions,[5] it is by far the largest source of variety in relation labels.
- **YAGO** (Tanon et al., 2020) is a KG which builds upon factual knowledge from Wikidata, but uses a limited set of 116 pre-defined relations from `schema.org` (Guha et al., 2016) mapped to a subset of Wikidata relations.
- **DBPedia** (Lehmann et al., 2015) is a KG that maps Wikipedia infotables to a predefined ontology containing 1,355 relations, about 350 of which are accompanied by a description.

We query all KGs using their openly available endpoints to retrieve a list of relations in each KG. For each relation, we retrieve up to five *triples* that use this relation, and the relation *description*, i.e. a short explanatory text. If present, we also retrieve descriptions for the head and tail entities.

We apply a set of filtering heuristics, leaving out e.g. relations describing KG metadata or identification numbers.[6] In this way, we collect 7,334 triples with 1,716 relations in total. For the full description regarding the data retrieval, please refer to Appendix A.

3.2 Annotation Process

We collect human-written verbalizations for all input triples using Prolific.[7] We built a web interface in which the human annotators are shown a single triple t and asked to describe it in a single sentence. The annotators are encouraged to re-use the entities in their original form, but they are able to change the form if necessary. The annotators can also report noisy inputs. We employed 420 annotators in total, each of which annotated 20 examples. We set the average reward per hour according to the platform recommendations to £7.29 per hour and we accepted all the inputs which pass our built-in checks. See Appendix B for more details on the annotation process.

3.3 Postprocessing the Data

A considerable portion of the collected verbalizations contain typos and grammatical errors, misunderstood meaning of the relation, or extra information in the input. To ensure high quality of our data, we manually examined all crowdsourced examples and annotated them as *OK*, *noisy*, *corrupted* or *containing extra information*. Appendix C includes postprocessing details. In the rest of the paper, we only use the subset of our dataset

[3] We did not use DART (which is a compilation of several datasets including WebNLG) for our experiments since it contains many noisy relations.

[4] https://www.w3.org/TR/PR-rdf-syntax/

[5] https://www.wikidata.org/wiki/Wikidata:Database_reports/List_of_properties/all

[6] Relations describing various IDs make up a suprisingly large portion of relations in Wikidata. Since we focus on diversity instead of coverage, we decided not to include these relations in our dataset.

[7] https://www.prolific.co/

with *OK* annotations, one per input triple (4,097 examples, 1,522 distinct relations), although we also make the remaining noisy instances available for future research.

4 Analysis and Evaluation

In our analysis, we are interested in the following research questions:

- **RQ1:** Are the PLMs finetuned for D2T generation able to describe relations for which they *do not have any parallel data*?
- **RQ2:** How many *training examples* do the PLMs need to generate satisfactory outputs?
- **RQ3:** How do the PLMs behave when provided *limited lexical cues* about the relation?
- **RQ4:** Can relation *descriptions* help to clarify ambiguous cases and improve semantic accuracy of the outputs?

To answer these questions, we divide our REL2TEXT dataset into a training and test splits (see §4.1 for details). We then use the **REL2TEXT test set** to evaluate a finetuned BART model (Lewis et al., 2020), a pretrained encoder-decoder transformer, which is used as a backbone of many recent data-to-text models (Ke et al., 2021; Xing and Wan, 2021; Ribeiro et al., 2020a; Liu et al., 2021).[8]

To answer *RQ1*, we compare the performance of BART finetuned on the REL2TEXT training set with BART finetuned on two qualitatively different D2T datasets – WEBNLG and KELM. Using REL2TEXT only, we then prepare various setups for answering *RQ2*, *RQ3*, and *RQ4* (details in §4.2). We analyze the outputs of the models both automatically (§4.3) and manually (§4.4).

4.1 Experimental Setup

Datasets We experiment with the following datasets, all of which focus on verbalizing factual information from KGs and use the same triple-based input data format:

- REL2TEXT. Our dataset (cf. §3.2) with single triples from three KGs with 4,097 examples, 1,522 relations and *human-annotated* outputs.
- WEBNLG (Ferreira et al., 2020; Gardent et al., 2017b). A DBPedia-based triple-to-text dataset with 38k examples, 411 relations, up to 7 triples per example, and *human-annotated* outputs. We use the English part of version 3.0 from HuggingFace.[9]
- KELM (Agarwal et al., 2021). A Wikidata-based dataset with 11M examples, 1,519 relations, up to 13 triples per example, and *model-generated* outputs. We use the dataset released by the authors, splitting it in a 1:100 ratio into validation and training data.

Rel2Text Data Split We use approximately 15% of the REL2TEXT examples for the **test set**. To ensure maximum fairness and focus on model generalization to unseen relations, we do not include in the REL2TEXT test set any relations which have an exact string match with a relation in KELM, WEBNLG, or the REL2TEXT training set. We also exclude any relations for which the maximum semantic similarity[10] to any KELM/WEBNLG/REL2TEXT training relation exceeds a threshold of 0.9. We set this threshold empirically in order to exclude relations which are almost synonymous, but slightly lexically different. We use 90% of the remaining examples for the training set and 10% for the validation set.

Data Preprocessing We split the camel-case in the relation labels. For finetuning the models, we linearize the input triples by marking the triple constituents with special tokens <*head*>, <*rel*> and <*tail*>, which we add to the model vocabulary.

Training and Decoding Setup In a default scenario, we finetune BART-BASE for 10 epochs and select the best checkpoint using validation BLEU score, then use greedy decoding to produce outputs. We repeat each experiment with five random seeds, averaging the results. See Appendix D for details.

4.2 Compared Systems

Copy Baseline We introduce a simple baseline by outputting the triple constituents separated by space: "e_h r e_t".

Full Training Data We use the default setup (§4.1) on full REL2TEXT and WEBNLG training sets. For KELM (which is about 300× larger than WebNLG), we finetune the model for 1 epoch only. We denote the trained models *full-rel2text*, *full-webnlg*, and *full-kelm*, respectively.

[8] We believe that our findings also apply to similar models such as T5 (Raffel et al., 2020), which have shown comparable performance on related tasks.

[9] `https://huggingface.co/datasets/web_nlg`

[10] Computed as cosine similarity between embeddings of the labels, which are encoded using `all-distilroberta-v1` from SBERT (Reimers and Gurevych, 2019).

Limited Training Data For the limited training data setup, we prepare few-shot splits from REL2TEXT as subsets containing $N = \{25, 50, 100, 200\}$ relations with a single example per relation. We select examples at random, ensuring that each few-shot split is a subset of the larger splits. We finetune the *fewshot-N* models for 10 epochs without validation, using the last checkpoint.

Limited Lexical Cues In D2T datasets, unclear relation labels are commonly reformulated, assuming that the models would otherwise not be able to verbalize them correctly (Gardent et al., 2017a). We investigate how the models behave if we take this issue to the extreme, i.e. if the relation labels are not available at all. We consider three scenarios:

- *mask-test* – We train the model on REL2TEXT in the standard training setup. For testing, we replace the relation labels in REL2TEXT with the <*mask*> token.
- *mask-train* – For training, we replace the relation labels in REL2TEXT with the <*mask*> token. We test the model on REL2TEXT in the standard evaluation setup.
- *mask-all* – We replace the relation labels in REL2TEXT with the <*mask*> token for both training and testing.

Incorporating Descriptions Our dataset contains short text descriptions of the relations, which may be useful to disambiguate its meaning and provide additional clues to the model. We consider two scenarios:

- *desc-repl* – We replace the relation label with its description.
- *desc-cat* – We concatenate the relation description with the input, separated using the special token <*rel_desc*>.

4.3 Automatic Evaluation

To get a high-level overview of model behavior, we evaluate generated outputs using the GEM-metrics[11] package (Gehrmann et al., 2021), which provides an extensive set of automatic metrics for text generation.

Lexical Similarity We first measure lexical similarity between the model outputs and human references using **BLEU** (Papineni et al., 2002), **METEOR** (Banerjee and Lavie, 2005), and **BLEURT** (Sellam et al., 2020). The first two metrics focus on n-gram overlap; the latter is a trained metric which also captures semantic similarity between the output and the reference. Although these metrics should not be used in isolation (Gehrmann et al., 2022), they give us a better overview of the output quality in combination with other metrics.

Semantic Similarity and Legibility Lexical similarity metrics focus on the surface form, which may not be telling the whole story. For example, if the relation *parent* denotes that e_t *is the parent of* e_h, but the entities are swapped in the generated text, the output will be incorrect, although lexical similarity metrics will be high. To get deeper insights into semantic and lexical properties of the outputs, we use NUBIA (Kane et al., 2020), which is a trained metric combining several features to measure "interchangeability" (equivalence) of two texts. The metric outputs a single score (**NB**) with a value between 0 and 1. We also report its individual underlying features: the semantic similarity score (**SS**) on a 0-5 scale, predicted by RoBERTa (Liu et al., 2019) finetuned on the STS-B benchmark (Cer et al., 2017); the contradiction (**C**), neutral (**N**), and entailment (**E**) probabilities from RoBERTa finetuned on the MNLI challenge from the GLUE benchmark (Wang et al., 2018); and the perplexity score (**PPL**) from vanilla GPT-2 (Radford et al., 2019), computed as a geometric mean of probabilities of the tokens in each step (this score is referenceless).

Lexical Diversity To assess lexical diversity of the generated texts, we use several metrics used in previous work (Dušek et al., 2020; van Miltenburg et al., 2018). We measure the number of unique n-grams (**U-1**), conditional entropy of bi-grams (**CE-2**), and the mean segmental type-token ratio over segment lengths of 100 (**MSTTR**; Johnson, 1944). We also measure the average output length in tokens (**len**).

4.4 Manual Error Analysis

To examine the sources of errors, we perform an in-house annotation of the model outputs. First, we identify four model error types based on preliminary observations of the model: semantic errors (SEM), with a swap of the relation direction

[11] https://github.com/GEM-benchmark/GEM-metrics

	Lexical			Semantics					Referenceless				
	BLEU	**METEOR**	**BLEURT**	**SS**	**C**	**N**	**E**	**NB**	**U-1**	**CE-2**	**MSTTR**	**PPL**	**len**
human	-	-	-	-	-	-	-	-	1785	2.13	0.62	5.88	9.55
copy	29.04	37.52	0.09	4.79	1.22	7.57	91.21	0.74	1606	1.17	0.7	7.55	6.72
full-rel2text	52.54	44.86	0.54	4.72	3.50	4.65	91.85	0.88	1661	1.96	0.58	5.89	9.16
full-webnlg	41.99	41.59	0.41	4.65	3.68	6.93	89.39	0.86	1651	2.54	0.56	5.65	10.29
full-kelm	46.74	42.94	0.46	4.70	3.95	5.29	90.77	0.86	1652	2.32	0.56	5.83	9.71
fewshot-25	31.13	35.52	-0.02	3.94	8.35	27.26	64.39	0.65	1445	2.93	0.52	5.34	10.67
fewshot-50	40.60	40.05	0.25	4.44	8.04	13.12	78.84	0.76	1536	2.31	0.55	5.79	9.90
fewshot-100	45.88	42.38	0.38	4.53	6.34	10.60	83.06	0.81	1600	2.13	0.57	5.85	9.57
fewshot-200	48.67	43.34	0.44	4.58	5.40	9.03	85.57	0.83	1626	2.04	0.58	5.89	9.36
mask-test	42.45	38.52	0.25	3.99	14.91	18.47	66.62	0.65	1669	1.96	0.61	5.69	8.96
mask-train	46.90	43.15	0.43	4.55	5.85	11.55	82.61	0.81	1646	2.00	0.57	5.91	9.74
mask-all	42.53	38.49	0.24	3.85	17.58	25.15	57.26	0.61	1677	1.96	0.61	5.66	9.16
desc-repl	49.35	42.85	0.47	4.57	5.78	8.80	85.42	0.82	1693	1.94	0.59	5.86	9.18
desc-cat	53.07	45.04	0.55	4.72	3.46	4.66	91.88	0.87	1668	1.91	0.59	5.92	9.11

Table 2: The summary of evaluation using automatic metrics on Rel2Text test set. See §4.1 for the descriptions of the models, §4.3 for the descriptions of the metrics, and Table 6 for standard deviations.

	label	**example input**	**model**	**example outputs (✗ incorrect, ✓ correct)**
model	Sem	(Yousra Matine, *sport country*, Morocco)	*mask-mask* *full-rel2text*	✗ Yousra Matine was born in Morocco. ✓ Yousra Matine plays for Morocco.
	Dir	(Kentucky Channel, *former broadcast network*, KET ED)	*fewshot-100* *full-rel2text*	✗ KET ED was broadcast on Kentucky Channel ED. ✓ The Kentucky Channel was broadcast on KET ED.
	Lit	(Vietnam Television, *first air date*, 1970-09-07)	*full-kelm* *full-rel2text*	✗ The first air date of Vietnam Television was 1970-09-07. ✓ Vietnam Television first aired on 1970-09-07.
	Lex	(RPG-43, *used in war*, The Troubles)	*full-rel2text* *full-kelm*	✗ RPG-43 was used in the The Troubles. ✓ The RPG-43 was used in the Troubles.
data	Ent	(The Age of Entitlement, *by artist*, The Basics)	*full-kelm* *full-rel2text*	✗ The Age of Entitlement was written by The Basics. ✓ The Age of Entitlement was recorded by The Basics.
	Lbl	(General Motors Epsilon platform, *vehicle*, Cadillac XTS)	*full-webnlg* *desc-cat*	✗ General Motors Epsilon is a vehicle similar to the Cadillac XTS. ✓ General Motors Epsilon platform is used in the Cadillac XTS.

Table 3: Error categories used in manual analysis, with examples of errors found and corresponding correct verbalizations. Model error types (top): Sem – The output is semantically incorrect, Dir – The direction of the relation is swapped, Lit – The verbalization is too literal, Lex – There is a lexical error in the output. Input data error types (bottom): Ent – The verbalization may depend on the entities, Lbl – The relation label is not clear.

(Dir) as a special case, too literal (Lit), i.e. containing awkward or misleading phrasing, and grammar/lexical errors (Lex). We further annotate 2 types of input data errors: ambiguous relations (Ent) and relations with unclear labels (Lbl). Table 3 shows examples for all categories. We select 100 random examples together with their corresponding outputs from the *full-rel2text*, *full-webnlg*, *full-kelm*, *fewshot-100*, *mask-all* and *desc-cat* models. Without revealing the output sources, we ask three expert annotators to mark all error categories that apply.

5 Results

5.1 Automatic Evaluation Results

Table 2 shows automatic scores for all our models. *full-rel2text* is the best among the fully trained models in terms of lexical overlap metrics (which is expected, as it trained on the most similar reference distribution), but the *full-webnlg* and *full-kelm* models are almost equal in terms of semantic consistency, achieving around 90% average entailment probability, which is on par with the copy baseline.

Semantic consistency is much lower for the few-shot models (e.g. the average entailment probability is between 65% and 85%), showing that there is a certain minimum amount of data needed to achieve consistent outputs. Interestingly, the models which do not see the relations during test time (*mask-test* and *mask-all*) still achieve around 60% average entailment probability, similar to the worst few-shot model. Although their rate of contradictions is higher than for other models, the results suggest that in many cases, the relation can be guessed from the related entities.

Another interesting observation is that the *mask-train* model (trained not to use the labels) is able to use the labels provided at test time to improve the outputs considerably (drop from 17% to 5% in contradiction rate compared to *mask-all*). The fact that relation labels carry most of the information for the successful verbalization is emphasized by the finding that the *desc-repl* model is worse than *full-rel2text* (although the descriptions are longer and supposedly explain the relation semantics), and the benefits of concatenating the descriptions alongside the relation labels (*desc-cat*) are negligible, only slightly improving lexical similarity metrics (0.5 BLEU point gain over *full-rel2text*).

In terms of lexical diversity, human references use more unique n-grams, but the model outputs are very similar in other aspects. It remains to be seen if the model outputs can stay semantically consistent with diversity-focused decoding techniques such as nucleus sampling (Holtzman et al., 2020).

5.2 Error Analysis Results

Results are summarized in Figure 2; complete results are presented in Appendix F. Table 3 shows examples of model outputs for each error type; more examples are given in Appendix G.

The *full-kelm* and *full-webnlg* models use expressions which are too literal (LIT) in 23 and 29 cases, respectively, while the *full-rel2text* and *desc-cat* models do the same only in 11 cases (five out of which are marked as LBL, i.e., with an unclear label). This suggests that the variability of our dataset helps the models to apply more natural expressions, especially if the relation is understandable from its label.

There is a near-constant portion of examples where the models make a semantic error (SEM) *and* the input is marked as needing an extra description (LBL). The models also make relatively many semantic errors on their own, most prominently in the case of the *fewshot-100* and the *mask-all* models. The *mask-all* model made a semantic error in 78 cases, suggesting that guessing the relation just from the entities is difficult (although still possible in 22 cases). Morevover, the outcomes from this model are fluent (only 4 LEX errors), making it hard to detect faulty cases.

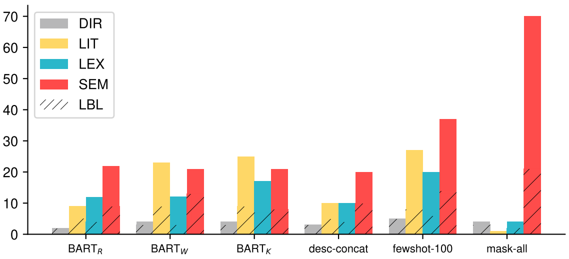


Figure 2: Number of annotated errors per model (see §4.4 and Table 3 for the description of error categories and §4.2 for the models). The striped part signifies that the label of the input was marked as unclear. See Appendix F for details.

The case of swapping the relation direction (DIR) is surprisingly not that common. This is probably down to having only a few examples in our dataset prone to this kind of error. Notably, the results for *full-rel2text* and *desc-cat* are very similar, rendering the impact of extra descriptions negligible.

Finally, there were only 12 out of 100 examples annotated as ENT, which suggests that the relation can be mostly decided irrespective of the entities in the triple.

6 Downstream Tasks

Given that the *full-rel2text* model can describe relations from their labels with high accuracy, we investigate if we can use the model to replace manually created templates in downstream tasks. We select two qualitatively different tasks, both using the idea of transforming individual input triples to simple sentences as a preprocessing step: tabular reasoning (§6.1) and zero-shot data-to-text generation (§6.2).

6.1 Tabular Reasoning

Gupta et al. (2020) presented the INFOTABS dataset as an NLI benchmark on tabular data. Each example is a structured table with a set of premises, i.e. natural language claims about the table; the task is to determine whether each premise is entailed by the table, contradicted by it, or neither.

They represent the table as *a paragraph* where each table cell is represented as a short sentence, mostly using a simple template "The *key* of *title* are *value*." Neeraja et al. (2021) extend Gupta et al.'s approach, including a *better paragraph representation* for which they prepare a

premise repr.	dev	α_1	α_2	α_3
OPR (Gupta et al., 2020)	76.78	75.30	68.46	64.63
BPR (Neeraja et al., 2021)	77.04	74.44	67.46	63.17
full-rel2text (ours)	74.44	74.31	64.59	63.46

Table 4: Accuracy for the dev set and test sets $\alpha_{1,2,3}$ from the INFOTABS dataset. The results are averaged over 3 random seeds.

dataset	model	BLEU	METEOR	O	H
filtered	orig	43.19	39.13	0.152	0.073
	full-rel2text	45.39	38.97	0.056	0.161
full	orig	42.92	39.07	0.051	0.148
	full-rel2text	44.63	38.93	0.058	0.166

Table 5: Lexical similarity metrics (BLEU, METEOR) and ommission (O) and hallucinaton (H) rate; following the setup in Kasner and Dušek (2022).

fine-grained set of rules for individual entity categories. The rules[12] aim to minimize the number of ungrammatical sentences and improve the reasoning abilities of the NLI model.

We replicate the setup of Neeraja et al. (2021) for the original (OPR) and better (BPR) paragraph representation using their public codebase. We then replace their templates with our *full-rel2text* model, verbalizing the triple (*title*, *key*, *value*). The results are summarized in Table 4.

Our preliminary manual evaluation suggests that the sentences from our model are indeed more grammatical (even compared to BPR). However, we observe that the performance is comparable across all three test sets. We hypothesize that although using the pretrained model can save us the effort in hand-crafting templates, the consistent format of the input appears to be more important than its fluency for classification tasks such as NLI.

6.2 Zero-shot Data-to-Text Generation

Kasner and Dušek (2022) proposed a setup for zero-shot D2T generation in which pretrained models are used to gradually transform text into the final description. The first step of the pipeline requires transforming individual triples into text. We focus on the WebNLG dataset, for which the authors manually created 354 templates.[13] We replicate the authors' setup using their public code, applying *full-rel2text* instead of the templates. The results are summarized in Table 5.

We note that the pipeline using our model for preprocessing is able to achieve improvements of $\sim$2 BLEU points, at the cost of a slightly higher omission and hallucination rate, but crucially without needing the manual effort to create templates. Preliminary examination shows that sentences produced by our model are qualitatively similar to the manual templates, but more varied. Unlike the templates, our model may verbalize a relation differently depending on the context. Overall, we argue that training a PLM on verbalizing individual relations can potentially replace the manual effort of creating simple templates, which will have a notable impact for scaling similar approaches to larger datasets.

7 Final Remarks

We analyzed the abilities of PLMs to verbalize unseen relations in KGs using their short labels. Based on our findings, we believe that having expressive data labels is essential for seamless adaptation of D2T systems to new domains. For datasets which do not follow standard naming conventions, such as for the Rotowire dataset with basketball summaries (Wiseman et al., 2017) which uses abbreviations for column headers (e.g. FG3A stands for *"the number of shots the player attempted beyond the arc"*), we argue that straightforward rephrasing of these labels to natural language may increase the robustness of D2T systems trained on this data.

We also showed that including relation descriptions on the input may help, albeit slightly. To achieve more notable improvements in output accuracy, it may be necessary to combine a more detailed specification regarding the relation direction, type, acceptable values, etc., together with a model able to reason about this specification.

The REL2TEXT dataset, which we collected for our analysis, can also be used for training the models for replacing manual templates while preserving or improving performance on downstream tasks.

The remaining open question is how to handle input data which is underspecified and may introduce ambiguities in the descriptions. We believe that detecting these cases in the input data and fixing them prior to generation (for example with knowledge-augmented systems or a human-in-the-loop setup) could be an interesting line of future work.

[12] Formalized using more than 250 lines of Python code: https://github.com/utahnlp/knowledge_infotabs/blob/main/scripts/preprocess/bpr.py#L120

[13] Available at https://github.com/kasnerz/zeroshot-d2t-pipeline/blob/main/templates/templates-webnlg.json

Limitations

Our analysis is limited to verbalizing single triples, which is only a stepping stone towards full-fledged G2T generation. To generate data for entire subgraphs, other issues need to be solved first, including compositional generalization and structure-aware modeling. Nevertheless, we believe that this simplified setting allows us to distill insights which are still applicable to G2T generation in general.

The factuality of the REL2TEXT dataset is tightly related to the data in the input KGs, which may contain outdated or incorrect information, and may be influenced by our processing methods (see Appendix A for details). Using the models trained on our dataset should be done with caution, since it can lead in producing harmful, imprecise, or factually incorrect statements.

We focus only on the English part of the KGs and English datasets. In the future, our approach could be extended to multilingual setting using multilingual PLMs and non-English parts of KGs. For more morphologically rich languages, an extra effort would have to be put into correctly inflecting the entities in the generated text.

Ethics Statement

As we are aiming to develop D2T systems which can robustly generate text for multiple domains, we are building upon PLMs which are known to reflect or amplify biases found in their pretraining corpus (Bender et al., 2021). Although the purpose of our study is to minimize these biases, the outputs of our models can still contain statements which are not aligned with the input data and user needs.

We collected our training and evaluation data through the Prolific crowdsourcing platform. We ensured that all the annotators were given an average reward per hour according to the platform recommendations and we put extra attention into informing the participants about the content and purpose of our study. We also manually filtered the output to minimize the amount of noisy references in our dataset. See 3.2 and Appendix B for more details on the annotation process.

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, NAACL-HLT 2021*, pages 3554–3565, Online.

Satanjeev Banerjee and Alon Lavie. 2005. METEOR: An Automatic Metric for MT Evaluation with Improved Correlation with Human Judgments. In *Proceedings of the Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization@ACL 2005*, pages 65–72, Ann Arbor, Michigan, USA.

Emily M. Bender, Timnit Gebru, Angelina McMillan-Major, and Shmargaret Shmitchell. 2021. On the Dangers of Stochastic Parrots: Can Language Models Be Too Big? In *FAccT '21: 2021 ACM Conference on Fairness, Accountability, and Transparency*, pages 610–623, Virtual Event / Toronto, Canada.

Nadjet Bouayad-Agha, Gerard Casamayor, Simon Mille, Marco Rospocher, Horacio Saggion, Luciano Serafini, and Leo Wanner. 2012. From Ontology to NL: Generation of Multilingual User-Oriented Environmental Reports. In *Natural Language Processing and Information Systems - 17th International Conference on Applications of Natural Language to Information Systems, NLDB 2012Proceedings*, volume 7337 of *Lecture Notes in Computer Science*, pages 216–221, Groningen, The Netherlands.

Daniel Cer, Mona Diab, Eneko Agirre, Iñigo Lopez-Gazpio, and Lucia Specia. 2017. SemEval-2017 Task 1: Semantic Textual Similarity Multilingual and Crosslingual Focused Evaluation. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, pages 1–14, Vancouver, Canada.

Philipp Cimiano, Janna Lüker, David Nagel, and Christina Unger. 2013. Exploiting Ontology Lexica for Generating Natural Language Texts from RDF Data. In *ENLG 2013 - Proceedings of the 14th European Workshop on Natural Language Generation, August 8-9, 2013*, pages 10–19, Sofia, Bulgaria.

Ondrej Dušek, David M. Howcroft, and Verena Rieser. 2019. Semantic Noise Matters for Neural Natural Language Generation. In *Proceedings of the 12th International Conference on Natural Language Generation, INLG 2019*, pages 421–426, Tokyo, Japan.

Ondrej Dušek and Zdeněk Kasner. 2020. Evaluating Semantic Accuracy of Data-to-Text Generation with Natural Language Inference. In *Proceedings of the 13th International Conference on Natural Language Generation, INLG 2020*, pages 131–137, Dublin, Ireland.

Ondrej Dušek, Jekaterina Novikova, and Verena Rieser. 2020. Evaluating the state-of-the-art of End-to-End Natural Language Generation: The E2E NLG challenge. *Comput. Speech Lang.*, 59:123–156.

WA Falcon et al. 2019. PyTorch Lightning. *GitHub. Note: https://github.com/PyTorchLightning/pytorch-lightning*, 3.

Michael Färber, Frederic Bartscherer, Carsten Menne, and Achim Rettinger. 2018. Linked data quality of DBpedia, Freebase, OpenCyc, Wikidata, and YAGO. *Semantic Web*, 9(1):77–129.

Thiago 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+)*.

Claire Gardent, Anastasia Shimorina, Shashi Narayan, and Laura Perez-Beltrachini. 2017a. Creating Training Corpora for NLG Micro-Planning. In *55th annual meeting of the Association for Computational Linguistics (ACL)*.

Claire Gardent, Anastasia Shimorina, Shashi Narayan, and Laura Perez-Beltrachini. 2017b. The WebNLG Challenge: Generating Text from RDF Data. In *Proceedings of the 10th International Conference on Natural Language Generation, INLG 2017, Santiago de Compostela*, pages 124–133, Spain.

Sebastian Gehrmann, Tosin Adewumi, Karmanya Aggarwal, Pawan Sasanka Ammanamanchi, Anuoluwapo Aremu, Antoine Bosselut, Khyathi Raghavi Chandu, Miruna-Adriana Clinciu, Dipanjan Das, Kaustubh Dhole, et al. 2021. The GEM Benchmark: Natural Language Generation, Its Evaluation and Metrics. In *Proceedings of the 1st Workshop on Natural Language Generation, Evaluation, and Metrics (GEM 2021)*, pages 96–120.

Sebastian Gehrmann, Elizabeth Clark, and Thibault Sellam. 2022. Repairing the Cracked Foundation: A Survey of Obstacles in Evaluation Practices for Generated Text. *CoRR*, abs/2202.06935.

Ramanathan V. Guha, Dan Brickley, and Steve Macbeth. 2016. Schema.org: evolution of structured data on the web. *Commun. ACM*, 59(2):44–51.

Qipeng Guo, Zhijing Jin, Xipeng Qiu, Weinan Zhang, David Wipf, and Zheng Zhang. 2020. CycleGT: Unsupervised Graph-to-Text and Text-to-Graph Generation via Cycle Training. In *Proceedings of the 3rd International Workshop on Natural Language Generation from the Semantic Web (WebNLG+).*, pages 77–88.

Vivek Gupta, Maitrey Mehta, Pegah Nokhiz, and Vivek Srikumar. 2020. INFOTABS: Inference on Tables as Semi-structured Data. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020*, pages 2309–2324, Online.

Armin Haller, Axel Polleres, Daniil Dobriy, Nicolas Ferranti, and Sergio José Rodríguez Méndez. 2022. An Analysis of Links in Wikidata. In *The Semantic Web - 19th International Conference, ESWC 2022, Proceedings*, volume 13261 of *Lecture Notes in Computer Science*, pages 21–38, Hersonissos, Crete, Greece.

Hamza Harkous, Isabel Groves, and Amir Saffari. 2020. Have Your Text and Use It Too! End-to-End Neural Data-to-Text Generation with Semantic Fidelity. In *Proceedings of the 28th International Conference on Computational Linguistics, COLING 2020*, pages 2410–2424, Barcelona, Spain (Online.

Ari Holtzman, Jan Buys, Li Du, Maxwell Forbes, and Yejin Choi. 2020. The Curious Case of Neural Text Degeneration. In *8th International Conference on Learning Representations, ICLR 2020*, Addis Ababa, Ethiopia.

Wendell Johnson. 1944. Studies in Language Behavior: A Program of Research. *Psychological Monographs*, 56(2):1–15.

Mihir Kale and Abhinav Rastogi. 2020. Template Guided Text Generation for Task-Oriented Dialogue. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020*, pages 6505–6520, Online.

Hassan Kane, Muhammed Yusuf Kocyigit, Ali Abdalla, Pelkins Ajanoh, and Mohamed Coulibali. 2020. NUBIA: NeUral Based Interchangeability Assessor for Text Generation. In *Proceedings of the 1st Workshop on Evaluating NLG Evaluation*, pages 28–37.

Zdeněk Kasner and Ondrej Dušek. 2020. Data-to-Text Generation with Iterative Text Editing. In *Proceedings of the 13th International Conference on Natural Language Generation, INLG 2020*, pages 60–67, Dublin, Ireland.

Zdeněk Kasner and Ondrej Dušek. 2022. Neural Pipeline for Zero-Shot Data-to-Text Generation. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2022*, pages 3914–3932, Dublin, Ireland.

Pei Ke, Haozhe Ji, Yu Ran, Xin Cui, Liwei Wang, Linfeng Song, Xiaoyan Zhu, and Minlie Huang. 2021. JointGT: Graph-Text Joint Representation Learning for Text Generation from Knowledge Graphs. In *Findings of the Association for Computational Linguistics: ACL/IJCNLP 2021*, volume ACL/IJCNLP 2021 of *Findings of ACL*, pages 2526–2538, Online Event.

Moniba Keymanesh, Adrian Benton, and Mark Dredze. 2022. What Makes Data-to-Text Generation Hard for Pretrained Language Models? *CoRR*, abs/2205.11505.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A Method for Stochastic Optimization. In *3rd International Conference on Learning Representations, ICLR 2015Proceedings*, San Diego, CA, USA.

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, NAACL-HLT 2019, Volume 1 (Long and Short Papers)*, pages 2284–2293, Minneapolis, MN, USA.

Anirban Laha, Parag Jain, Abhijit Mishra, and Karthik Sankaranarayanan. 2019. Scalable Micro-planned Generation of Discourse from Structured Data. *Comput. Linguistics*, 45(4):737–763.

Chia-Hsuan Lee, Hao Cheng, and Mari Ostendorf. 2021. Dialogue State Tracking with a Language Model using Schema-Driven Prompting. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, EMNLP 2021*, pages 4937–4949, Virtual Event / Punta Cana, Dominican Republic.

Jens Lehmann, Robert Isele, Max Jakob, Anja Jentzsch, Dimitris Kontokostas, Pablo N. Mendes, Sebastian Hellmann, Mohamed Morsey, Patrick van Kleef, Sören Auer, and Christian Bizer. 2015. DBpedia - A large-scale, multilingual knowledge base extracted from Wikipedia. *Semantic Web*, 6(2):167–195.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. BART: Denoising Sequence-to-Sequence Pretraining for Natural Language Generation, Translation, and Comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020*, pages 7871–7880, Online.

Ye Liu, Yao Wan, Lifang He, Hao Peng, and Philip S. Yu. 2021. KG-BART: Knowledge Graph-Augmented BART for Generative Commonsense Reasoning. In *Thirty-Fifth AAAI Conference on Artificial Intelligence, AAAI 2021, Thirty-Third Conference on Innovative Applications of Artificial Intelligence, IAAI 2021, The Eleventh Symposium on Educational Advances in Artificial Intelligence, EAAI 2021*, pages 6418–6425, Virtual Event.

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. *CoRR*, abs/1907.11692.

Simon Mille, Kaustubh D. Dhole, Saad Mahamood, Laura Perez-Beltrachini, Varun Gangal, Mihir Kale, Emiel van Miltenburg, and Sebastian Gehrmann. 2021. Automatic Construction of Evaluation Suites for Natural Language Generation Datasets. In *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks 1, NeurIPS Datasets and Benchmarks 2021*.

Linyong Nan, Dragomir R. Radev, Rui Zhang, Amrit Rau, Abhinand Sivaprasad, Chiachun Hsieh, Xiangru Tang, Aadit Vyas, Neha Verma, Pranav Krishna, Yangxiaokang Liu, Nadia Irwanto, Jessica Pan, Faiaz Rahman, Ahmad Zaidi, Mutethia Mutuma, Yasin Tarabar, Ankit Gupta, Tao Yu, Yi Chern Tan, Xi Victoria Lin, Caiming Xiong, Richard Socher, and Nazneen Fatema Rajani. 2021. DART: Open-Domain Structured Data Record to Text Generation. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2021*, pages 432–447, Online.

J. Neeraja, Vivek Gupta, and Vivek Srikumar. 2021. Incorporating External Knowledge to Enhance Tabular Reasoning. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2021*, pages 2799–2809, Online.

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, July 6-12, 2002*, pages 311–318, Philadelphia, PA.

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. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020*, pages 1173–1186, Online.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Köpf, Edward Z. Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. 2019. PyTorch: An Imperative Style, High-Performance Deep Learning Library. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019*, pages 8024–8035, Vancouver, BC, Canada.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language Models Are Unsupervised Multitask Learners. Technical report, OpenAI.

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. *J. Mach. Learn. Res.*, 21:140:1–140:67.

Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP 2019*, pages 3980–3990, Hong Kong, China.

Leonardo F. R. Ribeiro, Martin Schmitt, Hinrich Schütze, and Iryna Gurevych. 2020a. Investigating Pretrained Language Models for Graph-to-Text Generation. *CoRR*, abs/2007.08426.

Leonardo F. R. Ribeiro, Yue Zhang, Claire Gardent, and Iryna Gurevych. 2020b. Modeling Global and Local Node Contexts for Text Generation from Knowledge Graphs. *Trans. Assoc. Comput. Linguistics*, 8:589–604.

Thibault Sellam, Dipanjan Das, and Ankur P. Parikh. 2020. BLEURT: Learning Robust Metrics for Text Generation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020*, pages 7881–7892, Online.

Xiantang Sun and Chris Mellish. 2006. Domain Independent Sentence Generation From RDF Representations for the Semantic Web. In *Combined Workshop on Language-Enabled Educational Technology and Development and Evaluation of Robust Spoken Dialogue Systems, European Conference on AI, Riva del Garda, Italy*.

Xiantang Sun and Chris Mellish. 2007. An Experiment on "free Generation" from Single RDF Triples. In *Proceedings of the Eleventh European Workshop on Natural Language Generation, ENLG 2007*, Schloss Dagstuhl, Germany.

Thomas Pellissier Tanon, Gerhard Weikum, and Fabian M. Suchanek. 2020. YAGO 4: A Reasonable Knowledge Base. In *The Semantic Web - 17th International Conference, ESWC 2020, Proceedings*, volume 12123 of *Lecture Notes in Computer Science*, pages 583–596, Heraklion, Crete, Greece.

Emiel van Miltenburg, Desmond Elliott, and Piek Vossen. 2018. Measuring the Diversity of Automatic Image Descriptions. In *Proceedings of the 27th International Conference on Computational Linguistics, COLING 2018*, pages 1730–1741, Santa Fe, New Mexico, USA.

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

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R. Bowman. 2018. GLUE: A Multi-Task Benchmark and Analysis Platform for Natural Language Understanding. In *Proceedings of the Workshop: Analyzing and Interpreting Neural Networks for NLP, BlackboxNLP@EMNLP 2018*, pages 353–355, Brussels, Belgium.

Xiaozhi Wang, Tianyu Gao, Zhaocheng Zhu, Zhengyan Zhang, Zhiyuan Liu, Juanzi Li, and Jian Tang. 2021. KEPLER: A Unified Model for Knowledge Embedding and Pre-trained Language Representation. *Trans. Assoc. Comput. Linguistics*, 9:176–194.

Sam Wiseman, Stuart M. Shieber, and Alexander M. Rush. 2017. Challenges in Data-to-Document Generation. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, EMNLP 2017*, pages 2253–2263, Copenhagen, Denmark.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, and Jamie Brew. 2019. HuggingFace's Transformers: State-of-the-art Natural Language Processing. *CoRR*, abs/1910.03771.

Xinyu Xing and Xiaojun Wan. 2021. Structure-Aware Pre-Training for Table-to-Text Generation. In *Findings of the Association for Computational Linguistics: ACL/IJCNLP 2021*, volume ACL/IJCNLP 2021 of *Findings of ACL*, pages 2273–2278, Online Event.

A Data Retrieval

DBPedia We query DBPedia through its SPARQL access point: `http://dbpedia.org/sparql`. We retrive relations as objects of type `rdf:Property` which have a property `rdfs:comment` (i.e., the relation description) and language `'en'`.

YAGO We download the English Wikipedia subset of YAGO 4 database dump from `https://yago-knowledge.org/downloads/yago-4`. We retrieve all objects of type `rdf:Property` which have a property `rdfs:comment`. For the entity descriptions, we parse the entity page at YAGO website `http://yago-knowledge.org/resource/`.

Wikidata We first use the Wikidata SPARQL access point: `https://query.wikidata.org/sparql` to retrieve the list of relations as objects of type `wikibase:Property` with `wikibase:language="en"`, together with their English descriptions (`lang(?altLabel) = "en"`). Second, we query Wikidata through the LDF endpoint `https://query.wikidata.org/bigdata/ldf`, which is better able to handle heavy requests, to retrieve the list of triples involved in the relation. Finally, for retrieving the entity descriptions, we use the API at https://www.wikidata.org/w/api.php.

Filtering We apply a comprehensive set of filters for filtering out noisy triples, including triples with entities containing meta-information (*"Category:"*, *"XMLSchema#"*), URLs, entites longer than 64 characters, relations having the string *"id"*, *"number"*, or *"code"* in the label, or having *"Reserved for DBpedia"* in the description. As a consequence, we lose some relations, most notably about 2/3 of the relations from Wikidata describing various identifiers (we opted for this step in order to maintain data diversity). If KGs contain relations with identical labels, we prefer the relations from DBPedia and YAGO (which have a substantially lower amount of relations) to Wikidata relations.

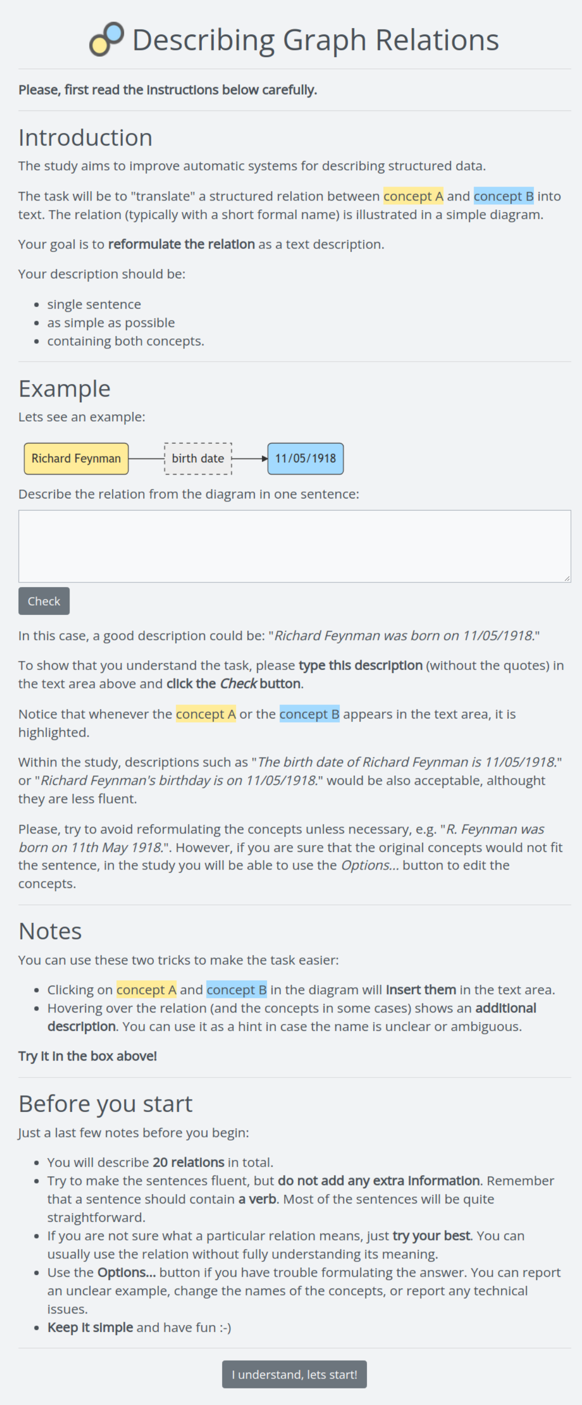
Missing Units Our dataset mostly does not contain units for quantities. Although the units are usually present in the KGs, they are not part of the quantity itself – they may be either connected to the quantity with another property, or described informally in the relation label. Since our focus was on the relation labels, we decided to not put additional effort in retrieving and processing the units. In effect, we consider verbalizations not using the units (e.g., (Bommersheim substation, *voltage*, 20000) → "Bommersheim substation has a voltage of 20000.") as correct.

Factual Correctness A certain part of the data is factually incorrect, either because there was an error in the knowledge graph (e.g., (Catalans, *population place*, **Italy**)) or because there was a processing error (e.g., (Child Language Teaching and Therapy, *final publication year*, **-1985**). Since our focus was not on judging the factuality of the inputs (which is a difficult problem on its own right), we decided to keep the examples in the dataset and consider the examples semantically consistent with the input triple as correct.

Other Notes

- All the data was retrieved in February 2022, except for YAGO where we used the newest available dump *2020-02-24*.
- Although we retrieved the entity descriptions wherever possible and we include them in our dataset, we decided not to use them in our experiments.
- The Python code for retriving the data is available in the paper repository.

B Crowdsourcing Details



Describing Graph Relations

Please, first read the instructions below carefully.

Introduction

The study aims to improve automatic systems for describing structured data.

The task will be to "translate" a structured relation between concept A and concept B into text. The relation (typically with a short formal name) is illustrated in a simple diagram.

Your goal is to **reformulate the relation** as a text description.

Your description should be:

- single sentence
- as simple as possible
- containing both concepts.

Example

Lets see an example:

Describe the relation from the diagram in one sentence:

In this case, a good description could be: "*Richard Feynman was born on 11/05/1918.*"

To show that you understand the task, please **type this description** (without the quotes) in the text area above and **click the *Check* button**.

Notice that whenever the concept A or the concept B appears in the text area, it is highlighted.

Within the study, descriptions such as "*The birth date of Richard Feynman is 11/05/1918.*" or "*Richard Feynman's birthday is on 11/05/1918.*" would be also acceptable, althought they are less fluent.

Please, try to avoid reformulating the concepts unless necessary, e.g. "*R. Feynman was born on 11th May 1918.*". However, if you are sure that the original concepts would not fit the sentence, in the study you will be able to use the *Options...* button to edit the concepts.

Notes

You can use these two tricks to make the task easier:

- Clicking on concept A and concept B in the diagram will **insert them** in the text area.
- Hovering over the relation (and the concepts in some cases) shows an **additional description**. You can use it as a hint in case the name is unclear or ambiguous.

Try it in the box above!

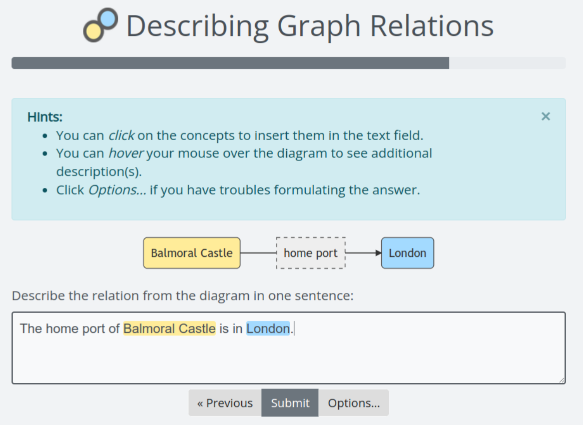
Before you start

Just a last few notes before you begin:

- You will describe **20 relations** in total.
- Try to make the sentences fluent, but **do not add any extra information**. Remember that a sentence should contain **a verb**. Most of the sentences will be quite straightforward.
- If you are not sure what a particular relation means, just **try your best**. You can usually use the relation without fully understanding its meaning.
- Use the **Options...** button if you have trouble formulating the answer. You can report an unclear example, change the names of the concepts, or report any technical issues.
- **Keep it simple** and have fun :-)

Figure 3: The introduction screen shown to the participants.

We built a web interface for collecting verbalizations for the triples. Figure 3 shows the introductory instructions displayed for the participants and Figure 4 shows the annotation interface.



Describing Graph Relations

Hints:
- You can *click* on the concepts to insert them in the text field.
- You can *hover* your mouse over the diagram to see additional description(s).
- Click *Options...* if you have troubles formulating the answer.

Describe the relation from the diagram in one sentence:

The home port of Balmoral Castle is in London.

Figure 4: The annotation inteface.

We hired annotators on the Prolific crowdsourcing platform https://app.prolific.co/. We required that the annotators are native speakers of English. After completing an introductory example, the annotators were given 20 randomly selected triples presented in a sequential order. The annotators were asked to write a short, single-sentence description of the triple. For making the annotation easier, hovering the mouse over the relation revealed its description (this applied also for the entities, if the description was present).

The annotators could also click on the entity to insert it in the text. This motivated the users to insert the entities in the original form. Once the entity appeared in the text (either typed or inserted), it was highlighted. We required that both entities (and at least two extra characters) are present in the text before proceeding to the next step. Because of this requirement, approximately 98.6% sentences in our dataset can be delexicalized using exact string matching. The users also had an option to modify the entity name, which would be recorded as a new ground-truth input (e.g., to make its form more natural). However, this option was used only sparingly.

In total, we collected 8,265 responses for 7,334 examples. Multiple responses for some examples are a consequence of random selection combined with sessions running in parallel. In the final dataset used in our experiments, we selected at most one correct answer for each example (see Appendix C).

C Postprocessing the Dataset

Two of the paper authors manually postprocessed the dataset. We used the following criteria for marking the responses:

- **OK** – The sentence is fluent and semantically consistent with the input.
- **Noisy** – The sentence contains a minor typographical or grammatical error, or the sentence sounds "awkward" (e.g., the relation label is used too literally).
- **Corrupted** – The sentence is semantically incorrect, contains a major typographical or grammatical error, or generally does not make sense.
- **Extra information** – The sentence is correct, but contains extra information about the entities which cannot be derived from the triple itself (e.g., the country of origin of the person found in the entity description).

Figure 5 shows the distribution of responses in our dataset. We marked 4,469 (54.1%) responses as *OK*, 1,314 (15.9%) responses as *Noisy*, 2,246 (27.2%) responses as *Corrupted* and 235 (2.8%) responses as *Extra information*.

Because our priority was to have clean data for evaluation, we decided to use only the *OK* part of our dataset in our experiments. We only use one example for each input triple in our experiments, which gives 4,097 instances. However, since we believe that the human outputs can also be an interesting research target, e.g. for investigating the feasibility of verbalizing the input data, we release all the annotations for future investigations.

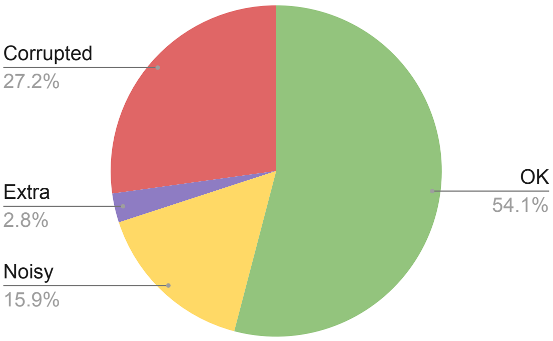


Figure 5: The distribution of crowdsourced responses in our dataset.

experiments	Lexical			Semantics					Referenceless				
	BLEU	METEOR	BLEURT	SS	C	N	E	NB	U-1	CE-2	MSTTR	PPL	len
full-rel2text	0.60	0.30	0.01	0.02	0.41	0.38	0.65	0.01	7	0.07	0.01	0.03	0.10
full-webnlg	0.69	0.09	0.00	0.02	0.23	0.94	1.07	0.00	7	0.02	0.00	0.02	0.10
full-kelm	0.78	0.22	0.01	0.02	0.49	0.15	0.42	0.01	11	0.03	0.00	0.03	0.06
fewshot-25	1.60	1.18	0.05	0.14	1.19	2.67	3.58	0.03	68	0.05	0.02	0.14	0.61
fewshot-50	1.36	0.59	0.02	0.06	0.99	0.77	1.59	0.02	19	0.13	0.01	0.08	0.19
fewshot-100	0.51	0.38	0.02	0.02	0.63	0.53	0.75	0.01	14	0.11	0.01	0.06	0.25
fewshot-200	0.80	0.35	0.02	0.02	0.60	1.37	1.25	0.01	14	0.06	0.00	0.02	0.08
mask-test	0.25	0.11	0.01	0.01	0.62	0.48	1.00	0.01	10	0.03	0.00	0.03	0.06
mask-train	0.19	0.09	0.01	0.03	0.64	1.73	1.95	0.01	10	0.02	0.01	0.03	0.09
mask-all	1.19	0.22	0.00	0.04	1.29	0.97	1.62	0.01	8	0.03	0.01	0.05	0.19
desc-repl	0.29	0.13	0.00	0.01	0.71	0.51	0.40	0.01	10	0.05	0.00	0.03	0.14
desc-cat	0.57	0.21	0.00	0.01	0.24	0.28	0.42	0.00	10	0.04	0.01	0.03	0.09

Table 6: Standard deviations for the model experiments in Table 2. Results were averaged across 5 random seeds.

D Experimental Setup

Framework We implemented the models in PyTorch Lightning (Paszke et al., 2019). We used the PyTorch (et al., 2019) version of BART-BASE from the Huggingface library (Wolf et al., 2019), with 140M parameters as a basis for all our models (except the copy baseline).

Hyperparameters We use the Adam (Kingma and Ba, 2015) optimizer ($\beta_1 = 0.9, \beta_2 = 0.98, \varepsilon = 1^{-6}$) with learning rate 2^{-5} and polynomial scheduling with 10% warmup steps. We train the models with batches of size 8 and accumulating gradients with factor 4 (an effective batch size of 32).

Training We train the models for 10 epochs on a single GeForce RTX 3090 GPU with 24 GB RAM, except for *full-kelm* model which we train for 1 epoch. Training times were around 15 minutes for the datasets based on REL2TEXT, 2 hours for *full-webnlg* and 3 days for *full-kelm*. We use greedy decoding in all our experiments.

E Automatic Evaluation

The standard deviations for each experiment from Table 2 are listed in Table 6.

F Manual Evaluation

Table 7 shows full results of our manual evaluation.

In total, there were 7 examples annotated as ENT only, 24 examples annotated as LBL only, and 5 examples annotated as ENT+DESC.

	DIR	LIT	LEX	SEM
full-rel2text	2 (0,2,0)	11 (1,5,1)	12 (0,4,0)	24 (2,9,1)
full-webnlg	8 (2,3,2)	23 (2,9,0)	12 (1,3,0)	25 (4,13,2)
full-kelm	6 (1,3,1)	29 (3,9,2)	19 (1,7,1)	25 (3,8,2)
fewshot-100	3 (0,3,0)	12 (2,5,1)	10 (0,4,0)	24 (2,10,2)
desc-cat	5 (0,5,0)	31 (3,8,2)	22 (2,6,1)	41 (6,14,2)
mask-all	4 (1,3,0)	1 (1,0,0)	4 (1,2,0)	78 (7,21,4)

Table 7: Number of errors manually annotated in 100 outputs from the models. The parentheses contain the portion of examples where the input was marked as ENT only, LBL only, and ENT+DESC, respectively.

G Examples

Examples of outputs of the models selected for manual evaluation are shown in Table 8.

input	(Kalatungan Mountain Range, *highest*, Mount Kalatungan)
desc.	*The highest mountain of a mountain range.*
full-rel2text	Kalatungan Mountain Range's highest mountain is Mount Kalatunan.
full-webnlg	The Kalatungan Mountain Range is the highest of Mount Kalatuna.
full-kelm	The highest mountain in the Kalatungan Mountain Range is Mount Kalatunagan.
desc-cat	Kalatungan Mountain Range is the highest mountain in the Kalatunan mountain range.
fewshot-100	Kalatungan Mountain Range is the highest mountain in the Kalatunan Mountain Range.
mask-all	Kalatungan Mountain Range is located on Mount Kalatunan.
input	(The Fight Before Christmas, *part of series*, The Simpsons)
desc.	*The series to which this episode or season belongs.*
full-rel2text	The Fight Before Christmas is part of The Simpsons.
full-webnlg	The Simpsons is the series that has The Fight Before Christmas as one of its episodes.
full-kelm	The Fight Before Christmas is part of the Simpsons series.
desc-cat	The Fight Before Christmas is a part of The Simpsons.
fewshot-100	The Fight Before Christmas was part of series of The Simpsons.
mask-all	The Fight Before Christmas was a parody of The Simpsons.
input	(Landwasser Viaduct, *construction material*, Limestone)
desc.	*Construction material (eg. concrete, steel, iron, stone, brick, wood).*
full-rel2text	Landwasser Viaduct was built using Limestone.
full-webnlg	The construction material of the Landwasser Viaduct is Limestone.
full-kelm	The Landwasser Viaduct is made of limestone.
desc-cat	Landwasser Viaduct was built using Limestone.
fewshot-100	Landwasser Viaduct was built using Limestone.
mask-all	Landwasser Viaduct crosses Limestone.
input	(Sharnbrook Academy, *county*, Bedfordshire)
desc.	*The county where the thing is located.*
full-rel2text	Sharnbrook Academy is in Bedfordshire.
full-webnlg	Sharnbrook Academy is in the county of Bedfordshire.
full-kelm	Sharnbrook Academy is located in Bedfordshire.
desc-cat	Sharnbrook Academy is in Bedfordshire.
fewshot-100	Sharnbrook Academy is in Bedfordshire.
mask-all	Sharnbrook Academy is in Bedfordshire.
input	(Loco-Motion, *computing platform*, Tomy Tutor)
desc.	*some sort of hardware architecture or software framework, that allows this software to run*
full-rel2text	Loco-Motion is run on Tomy Tutor.
full-webnlg	Tomy Tutor is the computing platform for Loco-Motion.
full-kelm	Loco-Motion is available for Tomy Tutor.
desc-cat	Loco-Motion runs on Tomy Tutor.
fewshot-100	Loco-Motion is a computing platform for Tomy Tutor.
mask-all	Loco-Motion was inspired by Tomy Tutor.

Table 8: Examples of outputs of the models selected for manual evaluation.