DCU-NLG-Small at the GEM'24 Data-to-Text Task: Rule-based generation and post-processing with T5-Base

Simon Mille, Mohammed Sabry and Anya Belz
ADAPT, Dublin City University
firstname.lastname@adaptcentre.ie

Abstract

Our submission to the GEM data-to-text shared task aims to assess the quality of texts produced by the combination of a rule-based system with a language model of reduced size. Our system first uses a rule-based generator to convert input triples into semantically correct English text, and then a language model to paraphrase these texts to make them more fluent. The texts are translated to languages other than English with the NLLB machine translation system.[1]

1 Introduction

On the one hand, Very Large Language Models are able to produce human-like texts from structured data but require enormous amounts of energy and computational resources to be trained, fine-tuned and run; on the other hand, resource-efficient techniques such as rule-based systems generally output texts that are less than optimally fluent. For our submission, we used three components: (i) a rule-based generator, FORGe (Mille et al., 2023b) to generate all inputs in English, (ii) a small-sized language model, T5-Base (Raffel et al., 2020), fine-tuned for rephrasing the rule-based outputs in a more fluent way, and (iii) an off-the-shelf Machine Translation system, NLLB (Team et al., 2022), for producing outputs in languages other than English. Our hypothesis is that using a language model for paraphrasing textual output produced by a reliable rule-based generator, rather than for directly mapping from triples to text, will make the system (i) more accurate in term of contents, i.e. less prone to omissions and additions (since all the contents of the input triples are already verbalised in the input of the language model), and (ii) generalise better to out-of-domain data, which represents five out of the six test sets of the GEM D2T task (since for the language model, instead of verbalising, the task is now paraphrasing, for which much more training data is available).

Input:

Subject	Property	Object
The_Haunted_Castle	imdbId	12
The_Haunted_Castle	director	Ezekiel_Kemboi
The_Haunted_Castle	director	Oleksandr_Turchynov

Possible English output:

Ezekiel Kemboi and Oleksandr Turchynov are the directors of The Haunted Castle, which has the IMDb identifier "12".

Figure 1: Sample GEM counterfactual input/output pair (D2T-1-CFA dataset).

In the remainder of the paper, we briefly summarise the GEM D2T shared task (Section 2), the rule-based generator and its extension (Section 3), the datasets we collected for fine-tuning T5 (Section 4), the fine-tuning procedure (Section 5), and the use of machine translation (Section 6); finally, we comment on the preliminary results (Section 7).

2 The GEM D2T Shared Task

In GEM D2T (Mille et al., 2024), the task is to generate texts in various languages starting from input triples extracted from DBpedia (Subtask 1) or Wikidata (Subtask 2) triples; see Figure 1 for an example of an input/output pair. Each subtask has three test sets: (i) a factual dataset (FA), which contains only factually correct information; (ii) a counterfactual dataset (CFA), which is the factual dataset but with entities (Subjects and Objects, see Figure 1) replaced by other entities of the same category (e.g. a person is replaced by another person, a date by another date, etc.); and (iii) a fictional dataset (FI), in which all Subject and Object values are fictional names made up by a language model.

The D2T-1 data is derived from WebNLG data (Castro Ferreira et al., 2020), while the D2T-

[1] Our code and data is available at https://github.com/dcu-nlg/GEM24-DCU-NLG-Small.

Proceedings of the 17th International Natural Language Generation Conference: Generation Challenges, pages 84–91
September 23 –27, 2024.

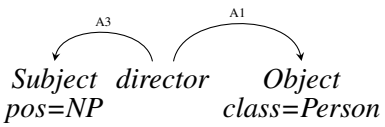

Figure 2: Sample PredArg template corresponding to the *director* property.

2 data was created for the present task using the method proposed by Axelsson and Skantze (2023) (i.e. collection of new Wikidata triples sets for a list of entities, and then replacement of entities according to steps (ii) and (iii) above). No training data was provided to the participants, and apart from the English Factual WebNLG data (i.e. the original test set in (Castro Ferreira et al., 2020)), no reference texts were available for any test set or language. The GEM organisers encouraged submissions in multiple languages, namely English (en), Chinese (zh), German (de), Russian (ru), Spanish (es), Korean (ko), Hindi (hi), Swahili (sw), and Arabic (ar), without saying beforehand which languages were going to be assessed.

3 Rule-based Generator

For our rule-based system, we use the FORGe generator (Mille et al., 2023b), which was partly developed on the WebNLG data. FORGe is implemented as a pipeline of modules that perform subtasks such as text planning, lexicalisation, sentence structuring and surface realisation. Each module consists of a set of rules (called *grammars*), which use dictionaries that describe the semantic and syntactic behaviours of the lexical units used in the verbalisations. The generator takes as input abstract predicate-argument structures manually crafted for each property, as shown in Figure 2.

FORGe already has such predicate-argument structures for the whole WebNLG 2020 dataset in English, which means that we were able to use FORGe off-the-shelf for Subtask 1; no modification was performed to address new entity names of the fictional test set. For Subtask 2, properties in the dataset built by the organisers come from the Wikidata vocabulary, which is different from the DBpedia vocabulary used in the WebNLG dataset. There are 74 different Wikidata properties, 17 of which have a direct mapping to a DBpedia property. For these 17 properties, we use the existing predicate-argument templates, while for the remaining 57 properties, new predicate-argument templates were crafted, referring to the Wikipedia pages of the entities used along each property to make sure we captured the correct semantics of each property. Crafting the 57 templates took approximately 2 hours. Minor updates to the generator's grammars were implemented to account for the specific aspects of the Wikidata test sets, in which the Subject is always the same, unlike in the WebNLG-based inputs.

4 Finetuning Datasets

Our objective in the paraphrasing component is to improve the fluency of the rule-based generator without sacrificing its semantic accuracy (i.e. avoiding what is commonly reported as omissions and hallucinations). For this, we collected parallel textual data, with on one side accurate but possibly disfluent texts (Text$_{Dis}$), and on the other side accurate and fluent texts (Text$_{Flu}$). In this section, we describe the three different datasets we created for the experiments; Section 5 reports on how we used this data for fine-tuning T5.

4.1 The forge2ref dataset

For data of type Text$_{Dis}$, we used texts generated with the FORGe rule-based system (see Section 3) as provided in the English version of the Mod-D2T dataset (Mille et al., 2023a),[2] which is a 10-layer version of the whole WebNLG 2020 dataset (training, development and test sets) produced with FORGe. For the parallel data of type Text$_{Flu}$, we used the corresponding list of reference texts from the original WebNLG 2020 data in each case, downloaded from HuggingFace.[3] The final data contains 13,211, 1,667, and 1,779 pairs in the training, development and test sets, respectively. The following is an example pair:

- Text$_{Dis}$: *The production of the Pontiac Rageous started in 1997. The Pontiac Rageous is a coupe.*
- Text$_{Flu}$: *['The Pontiac Rageous coupe went into production in 1997.', 'The Pontiac Rageous, first produced in 1997, was a car with a coupe body style.', 'The coupe style Pontiac Rageous was first produced in 1997.']*

4.2 The forge2llm dataset

In order to acquire additional high quality data, we also collected a very small set of language model

[2] https://github.com/mille-s/Mod-D2T/tree/main/conllu-en_INLG23

[3] https://huggingface.co/datasets/webnlg-challenge/web_nlg

outputs, using the best systems and the human evaluation results of the WebNLG 2020 shared task. Three systems competing in the 2020 edition of the shared task achieved human-level fluency: AmazonAI (Guo et al., 2020), FBConvAI (Yang et al., 2020) and OSU Neural NLG (Li et al., 2020). Assuming that these systems are generally able to output very fluent text, we selected the subset of these system outputs that were rated 0.95 or more when computing the mean for the three criteria related with the semantic faithfulness to the input triples, namely:

- "DataCoverage: *Does the text include descriptions of all predicates presented in the data?*;
- Relevance: *Does the text describe only such predicates (with related subjects and objects), which are found in the data?*;
- Correctness: *When describing predicates which are found in the data, does the text mention correct the objects and adequately introduces the subject for this specific predicate?*" (*sic*).

The system outputs and human ratings were obtained from the WebNLG GitHub repository.[4] For 163 inputs, we found between one and three system outputs that met our threshold (301 texts in total). These 163 lists of texts served as Text$_{Flu}$ data, and were paired with the corresponding FORGe texts serving as Text$_{Dis}$, e.g.:

- Text$_{Dis}$ (same as forge2ref's Text$_{Dis}$): *The production of the Pontiac Rageous started in 1997. The Pontiac Rageous is a coupe.*
- Text$_{Flu}$: *['The Pontiac Rageous has a Coupe body style and its production started in 1997.', 'Production of the Pontiac Rageous Coupe began in 1997.']*

Note that the data we are using for the forge2llm dataset constitutes about 9% of the D2T-1-FA test set (we use 163 data points out of the 1,779 data points in the test set). We thus expect this to slightly inflate our metrics scores on the D2T-1-FA set, but should not have an important impact on the other test sets.

4.3 The triple2ref dataset

For this dataset, we paired triples and human-written texts, both extracted from the WebNLG 2020 dataset (Castro Ferreira et al., 2020). The input triples are simply concatenated with a comma and a space, and the output reference texts are combined into a list. This dataset is used in addition to the other two for one of the models in order to increase its robustness to bad inputs. The final data contains 13,211, 1,667, and 1,779 pairs in the training, development and test sets respectively, e.g.:

- Text$_{Dis}$: *Pontiac_Rageous | productionStartYear | 1997, Pontiac_Rageous | bodyStyle | Coupe*
- Text$_{Flu}$ (same as forge2ref's Text$_{Flu}$): *['The Pontiac Rageous coupe went into production in 1997.', 'The Pontiac Rageous, first produced in 1997, was a car with a coupe body style.', 'The coupe style Pontiac Rageous was first produced in 1997.']*

5 Paraphrasing with T5-Base

In this section, we introduce T5-Base and all the models fine-tuned for our experiments.

5.1 T5: Experimental setup and model configuration

We conducted experiments with the T5-Base V1 model (250M parameters),[5] alongside one full-tuning technique and two parameter-efficient fine-tuning (PEFT) techniques, namely LoRA (Hu et al., 2021) and Adapter (Houlsby et al., 2019). The primary task was text-to-text generation, with the aim of transforming FORGe outputs into more fluent text. The T5-Base model does not inherently possess task-specific knowledge relevant to this task, but it is well-suited for text-to-text modelling tasks like paraphrasing.

For the evaluation phase, the model generation settings were as follows: Temperature: 0.1; Top-k: 100; Top-p: 0.95; Repetition penalty: 0.8.

5.2 Fine-tuning experiments

All models were trained using cross-entropy loss. For evaluating performance, we employed the HuggingFace Evaluate Library[6] to calculate the BLEU[7] and METEOR[8] metrics, comparing the predicted text against all available references for each input.

[4] https://github.com/WebNLG/challenge-2020

[5] https://huggingface.co/google/t5-v1_1-base

[6] https://huggingface.co/docs/evaluate/en/index

[7] https://huggingface.co/spaces/evaluate-metric/bleu

[8] https://huggingface.co/spaces/evaluate-metric/meteor

We used the training and development sets, keeping the test set for final model evaluation.

In our experiments, we tested three different fine-tuning techniques:

- **Full-tune** involves updating all parameters of a model to better suit a downstream task. This traditional method, while effective, becomes increasingly costly as model sizes scale up, prompting research into more parameter-efficient alternatives (Sabry and Belz, 2023).
- **Adapter** (Houlsby et al., 2019) is a parameter-efficient fine-tuning technique where a small set of trainable parameters, typically two linear layers with an activation function in between, is inserted at strategic locations within a model, such as after the attention and feed-forward blocks of a transformer model. Only these newly introduced parameters are updated during finetuning, while the original model's parameters remain fixed. We implemented Adapter with a bottleneck dimension of 64 and a GeLU activation function.
- **LoRA** (Hu et al., 2021) adopts a similar approach, adding a small set of trainable parameters; however, it specifically targets the query and key matrices within the attention blocks of transformers. These added parameters are viewed as a reparameterised form of the existing matrices, designed to accommodate task-specific adjustments without altering the original, fixed parameters of the model. We used LoRA Configuration of a rank of 8, alpha of 16, and a dropout rate of 0.0.

For each fine-tuning technique, we tested 4 sets of conditions involving different combinations of datasets from Section 4 to assess their performance, for a total of 12 different fine-tuned models:

- **10K**: Fine-tuning for 10,000 learning steps exclusively on forge2ref;
- **15K**: Fine-tuning for 15,000 learning steps on the two datasets that use FORGe texts, forge2ref and forge2llm;
- **35K**: Fine-tuning for 35,000 learning steps on all three datasets: forge2ref, forge2llm and triple2ref;
- **Avg.Prm**: Average of the trainable parameters from each of the 3 fine-tuning techniques (Full-tune, Adapters, LoRA, see above). This approach is based on findings that averaging multiple checkpoints can lead to better generalisation (Izmailov et al., 2018).

All models were trained using a learning rate of 6e-5, with a Cosine decay scheduler and 10% of the learning steps designated as a warm-up period. The training and evaluation batch sizes were set at 16.3K and 4K tokens, respectively. Additionally, a weight decay of 0.1 was implemented.

Training the T5-Base in NVIDIA A100-SXM-80GB for 10,000 steps with full precision (FP32), in our initial experiment, required the following GPU durations: 1 hour and 14 minutes for full fine-tuning, 55 minutes for Adapters, and 50 minutes for LoRA. This resulted in total computations of 86 petaFLOPs for full fine-tuning, 64 petaFLOPs for Adapters, and 58 petaFLOPs for LoRA, with corresponding energy consumptions of 0.493 kWh (kilowatt-hours), 0.367 kWh, and 0.333 kWh, respectively. Scaling the same settings to T5-Large could require roughly 3.5 times more, considering the difference in the parameters of the two models (0.2B for T5-Base vs. 0.7B for T5-Large).

When running the paraphrasing, each input is encoded in a maximum of 512 tokens, and the model is set to generate a maximum of 400 tokens. With an A100 GPU, T5 base (FP32) can process about 9K tokens per seconds, which means the paraphrasing time for one full test set (1,8K texts) is about 3 minutes.

6 Machine Translation with NLLB

The combination of resources we need for applying our approach (rule-based generator + parallel textual data) is currently only available in English. For producing outputs in other languages, we used the freely available NLLB machine translation tool (Team et al., 2022). NLLB is a pre-trained model that covers translation between numerous languages; it is available through HuggingFace[9] and can be executed on various types of runtimes, including CPUs. Each English text was split into sentences, and sentences were processed one at a time by NLLB; the translated sentences were then brought back together as a text and stored in the same format as the English outputs, in a .txt file with one text per line. We ran nllb-200-distilled-1.3B on a T4 GPU on Google Colab, which generally needed between 30 and 60 minutes to translate

[9] https://huggingface.co/docs/transformers/en/model_doc/nllb

Fine-tuning	Cond.	BLEU	METEOR
LoRA	10K	0.251	0.494
	15K	0.295	0.536
	35K	0.373	0.585
	Avg.Prm	0.305	0.544
Adapters	10K	0.480	0.670
	15K	0.476	0.671
	35K	0.508	0.694
	Avg.Prm	0.487	0.682
Full-tune	10K	0.506	0.702
	15K	0.542	0.721
	35K	0.536	0.719
	Avg.Prm	0.538	0.719

Table 1: BLEU and METEOR scores (with multiple references) of the 12 fine-tuned T5-Base models on the D2T-1-FA test set; see Section 5.2 for details about the fine-tuning techniques and the conditions.

a file. With 48 files to translate (8 target languages, 6 test sets per language) and several server interruptions, we had to finish the translations on a local HPC cluster to finish the translations on time.

We did not try to improve the translation quality, and did not perform any systematic qualitative analysis of the translated texts; for a few languages (Spanish, Hindi, German), we asked native speakers to browse through a few translations to have an idea of the general quality, which was judged sufficient to submit the outputs.

7 Results and Submitted Systems

In this section, we present the results of evaluating our 12 models on the D2T-1-FA test data, using the WebNLG 2020 reference texts for calculating BLEU (Papineni et al., 2002) and METEOR (Banerjee and Lavie, 2005), based on which we selected the model for our submission. We then briefly discuss the official results of the GEM D2T task as provided by the organisers, which at this point are restricted to the metrics scores for English (we report BLEU, METEOR and BertScore-F1 (Zhang et al., 2019)).

7.1 Own evaluation of the fine-tuned models

In Table 1, we report our own BLEU and METEOR scores on the English factual dataset of Subtask 1 (D2T-1-FA), the only one for which references are available at the time of writing.

For all systems, both metrics indicate the same tendencies: the **Full-tune** technique produces the highest scores, closely followed by **Adapter**. With **LoRA**, the results are much lower. We attribute this performance to the small set of parameters added by LoRA, the fact that they interact with the Attention block queries and keys, whereas the tasks require extensive manipulation of factual knowledge, stored in and retrieved from the FeedForward block (Geva et al., 2021). However, increasing the model size, carefully selecting hyperparameters, and/or extending the number of learning steps could mitigate these issues.

With **LoRA**, the more data, the better the results, while that is not necessarily the case for the other two techniques: **Adapter** produces very similar scores with (15K) and without (10K) the forge2llm data but gets better when adding the triple2ref data and learning steps (35K). **Full-tune** benefits more from the forge2llm data (15K) but not from adding the learning steps and the triple2ref data (35K). Since full fine-tuning involves adjusting a larger number of parameters, which allows for a greater degree of freedom to change, the model may initially focus on noisy signals before achieving convergence or being steered in the desired direction by the introduction of triple2ref data (intended to enhance model robustness). We suspect that the number of learning steps allocated may not be sufficient to accommodate these changes.

Finally, averaging the weights from the three fine-tuning techniques produces scores that are between those obtained for 15K and 35K learning steps, in terms of both BLEU and METEOR.

7.2 Submissions

We submitted the *Full-tune Avg.Prm* model, which did not obtain the absolute highest scores for both metrics, but which is supposed to be more robust to input variations (**Ours** in Table 2). As a secondary

System ID	BLEU	METEOR	Bert F1
System 2	52.26	0.410	0.956
Ours	**51.43**	**0.395**	**0.954**
System 4	51.36	0.410	0.955
System 1	49.8	0.400	0.955
System 5	43.09	0.389	0.950
System 6	42.38	0.390	0.946
Ours$_{NoT5}$	**40.55**	**0.372**	**0.943**
System 7	39.86	0.400	0.947
System 8	34.71	0.280	0.923

Table 2: Metrics evaluation of our *Full-tune Avg.Prm* system on the WebNLG 2020 test set provided by the organisers (sorted by BLEU score).

	D2T-1			D2T-2		
	FA	**CFA**	**FI**	**FA**	**CFA**	**FI**
BLEU	27.0	22.98	20.85	19.48	24.9	16.88
METEOR	0.314	0.279	0.292	0.26	0.3	0.267
chrF++	0.537	0.488	0.507	0.438	0.51	0.442
BERT F1	0.93	0.918	0.914	0.925	0.923	0.914

Table 3: Metrics scores for our DCU-NLG-Small submission for the English D2T task released by the organisers.

submission, and for comparison, we submitted all outputs of the rule-based generator without the T5 post-processing (**Ours**$_{NoT5}$ in Table 2). We also submitted outputs for all languages other than English, all produced by running NLLB off-the-shelf on the FORGe+T5 outputs.

7.3 GEM automatic evaluation results

Table 2 shows the first results released by the organisers, i.e. the metrics for the full English test set using all WebNLG 2020 reference texts (1,779 inputs, 2.5 reference texts per input on average). The scores of our system on this dataset cannot be clearly interpreted, since as mentioned in Section 4.2, we use a small portion of this dataset to fine-tune one of the models whose parameters were averaged to make the submitted model. One thing that can be noticed is the extent of the increase of the BLEU score when integrating the T5 post-processing. With close to 11 BLEU points difference, this suggests that our system outputs with T5 are much more similar to the reference texts than the FORGe outputs, which was expected.[10]

The organisers then released metrics results on the 6 D2T test sets (180 inputs each), using references collected on Amazon Mechanical Turk (one reference text per input); see Table 3. On the D2T-1 datasets, our system's scores substantially drop on the counterfactual (CFA) and fictional (FI) datasets; compared to the other participating systems, ours actually is the one that has the most substantial score decrease. In contrast, we have one of the smallest decreases between the factual (FA) and fictional (FI) D2T-2 datasets. Surprisingly, the D2T-2-CFA scores are higher than the D2T-1-CFA counterpart, and also than the D2T-2-FA score. However, all participating systems exhibited the same patterns, so it is likely that the data is somewhat responsible for this oddity. In general, the results of the human evaluation on the 6 test sets will shed more light on the actual quality of the contents produced by our system.

[10]Regarding the differences between the scores in Tables 1 and 2: for BLEU, there are differences between the evaluation package we used (Evaluate library from HuggingFace) and the commonly used WebNLG evaluation package, in particular in the smoothing factors applied in the BLEU metric calculation. This explains the 2.5-point discrepancy in BLEU scores observed between the results labelled 'Avg. Prm Full-tune' in Table 1 and 'Ours' in Table 2. In addition, for our own computation of METEOR, we used multiple references, as opposed to single references for the organisers, so the METEOR scores in Tables 1 and 2 are not comparable.

8 Conclusions

We have presented the DCU-NLG-Small submission to the GEM'24 Data-to-text shared task. Our system combines a rule-based generator that converts triples into English text, with a small language model that paraphrases the text to improve its fluency. An off-the-shelf MT system is used for producing outputs in the other languages. Our system performs better than a purely rule-based system according to metrics on an existing English test set, but generally undergoes substantial score decreases when confronted with different types of out-of-domain data. We hope that the human evaluation results will allow us to draw more definitive conclusions.

Acknowledgements

We would like to thank Michela Lorandi for her help with running the code to produce the machine-translated outputs. Mille's work was funded by the European Union under the Marie Skłodowska-Curie grant agreement No 101062572 (M-FleNS). Sabry's PhD is funded by the ADAPT SFI Centre for Digital Media Technology. Our work on this paper has also benefited more generally from being carried out within the ADAPT SFI Centre which is funded by Science Foundation Ireland through the SFI Research Centres Programme and is co-funded under the European Regional Development Fund (ERDF) through Grant 13/RC/2106.

References

Agnes Axelsson and Gabriel Skantze. 2023. Using large language models for zero-shot natural language

generation from knowledge graphs. *arXiv preprint arXiv:2307.07312*.

Satanjeev Banerjee and Alon Lavie. 2005. Meteor: An automatic metric for mt evaluation with improved correlation with human judgments. In *Proceedings of the acl workshop on intrinsic and extrinsic evaluation measures for machine translation and/or summarization*, pages 65–72.

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.

Mor Geva, Roei Schuster, Jonathan Berant, and Omer Levy. 2021. Transformer feed-forward layers are key-value memories. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 5484–5495, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Qipeng Guo, Zhijing Jin, Ning Dai, Xipeng Qiu, Xiangyang Xue, David Wipf, and Zheng Zhang. 2020. $\mathcal{P}^2$: A plan-and-pretrain approach for knowledge graph-to-text generation. In *Proceedings of the 3rd International Workshop on Natural Language Generation from the Semantic Web (WebNLG+)*, pages 100–106, Dublin, Ireland (Virtual). Association for Computational Linguistics.

Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin De Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. 2019. Parameter-efficient transfer learning for NLP. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 2790–2799. PMLR.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, and Weizhu Chen. 2021. Lora: Low-rank adaptation of large language models. *CoRR*, abs/2106.09685.

Pavel Izmailov, Dmitrii Podoprikhin, Timur Garipov, Dmitry P. Vetrov, and Andrew Gordon Wilson. 2018. Averaging weights leads to wider optima and better generalization. *CoRR*, abs/1803.05407.

Xintong Li, Aleksandre Maskharashvili, Symon Jory Stevens-Guille, and Michael White. 2020. Leveraging large pretrained models for WebNLG 2020. In *Proceedings of the 3rd International Workshop on Natural Language Generation from the Semantic Web (WebNLG+)*, pages 117–124, Dublin, Ireland (Virtual). Association for Computational Linguistics.

Simon Mille, Francois Lareau, Stamatia Dasiopoulou, and Anya Belz. 2023a. Mod-D2T: A multi-layer dataset for modular data-to-text generation. In *Proceedings of the 16th International Natural Language Generation Conference*, pages 455–466, Prague, Czechia. Association for Computational Linguistics.

Simon Mille, João Sedoc, Yixin Liu, Elizabeth Clark, Agnes Axelsson, Miruna-Adriana Clinciu, Yufang Hou, Saad Mahamood, Ishmael Obonyo, and Lining Zhang. 2024. The 2024 GEM shared task on multilingual data-to-text generation and summarization: Overview and preliminary results. In *Proceedings of the 17th International Conference on Natural Language Generation: Generation Challenges*, Tokyo, Japan. Association for Computational Linguistics.

Simon Mille, Elaine Uí Dhonnchadha, Lauren Cassidy, Brian Davis, Stamatia Dasiopoulou, and Anya Belz. 2023b. Generating Irish text with a flexible plug-and-play architecture. In *Proceedings of the 2nd Workshop on Pattern-based Approaches to NLP in the Age of Deep Learning*, pages 25–42, Singapore. 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.

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.

Mohammed Sabry and Anya Belz. 2023. Peft-ref: A modular reference architecture and typology for parameter-efficient finetuning techniques. *Preprint*, arXiv:2304.12410.

NLLB Team, Marta R. Costa-jussà, James Cross, Onur Çelebi, Maha Elbayad, Kenneth Heafield, Kevin Heffernan, Elahe Kalbassi, Janice Lam, Daniel Licht, Jean Maillard, Anna Sun, Skyler Wang, Guillaume Wenzek, Al Youngblood, Bapi Akula, Loic Barrault, Gabriel Mejia Gonzalez, Prangthip Hansanti, John Hoffman, Semarley Jarrett, Kaushik Ram Sadagopan, Dirk Rowe, Shannon Spruit, Chau Tran, Pierre Andrews, Necip Fazil Ayan, Shruti Bhosale, Sergey Edunov, Angela Fan, Cynthia Gao, Vedanuj Goswami, Francisco Guzmán, Philipp Koehn, Alexandre Mourachko, Christophe Ropers, Safiyyah Saleem, Holger Schwenk, and Jeff Wang. 2022. No language left behind: Scaling human-centered machine translation. *Preprint*, arXiv:2207.04672.

Zixiaofan Yang, Arash Einolghozati, Hakan Inan, Keith Diedrick, Angela Fan, Pinar Donmez, and Sonal Gupta. 2020. Improving text-to-text pre-trained models for the graph-to-text task. In *Proceedings of the 3rd International Workshop on Natural Language Generation from the Semantic Web (WebNLG+)*, pages 107–116, Dublin, Ireland (Virtual). Association for Computational Linguistics.

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