DCU-NLG-PBN at the GEM'24 Data-to-Text Task: Open-Source LLM PEFT-Tuning for Effective Data-to-Text Generation

Michela Lorandi and **Anya Belz**
ADAPT Research Centre, Dublin City University
{michela.lorandi, anya.belz}@adaptcentre.ie

Abstract

LLMs have been used in various tasks with impressive success, including data-to-text generation. However, one concern when LLMs are compared to alternative methods is data contamination, in other words, for many datasets the data used in training these models may have included publicly available test sets. In this paper, we explore the performance of LLMs using newly constructed datasets in the context of data-to-text generation for English, Chinese, German, Russian, Spanish, Korean, Hindi, Swahili, and Arabic. We performed a testing phase to evaluate a range of prompt types and a fine-tuning technique on Mistral 7B and Falcon 40B. We then fully evaluated the most promising system for each scenario: (i) LLM prompting in English followed by translation, and (ii) LLM PEFT-tuning in English followed by translation. We find that fine-tuning Mistral outperforms all other tested systems and achieves performance close to GPT-3.5. The few-shot prompting with a dynamic selection of examples achieves higher results among prompting. The human evaluation to be carried out by the shared-task organisers will provide insight into the performance of the new datasets. In conclusion, we observed how the fine-tuning of an open-source LLM can achieve good performance close to state-of-the-art closed-source LLM while using considerably fewer resources.

1 Introduction

With the advancement of Large Language Models (LLMs), their capabilities have been explored in many tasks including data-to-text generation, which maps structured input data into a suitable output text containing all and only provided information. However, the datasets for many data-to-text tasks have been available online for years and might have been used to train LLMs. In the work reported here, we participate in the GEM 2024 shared task (Mille et al., 2024) using new datasets which are not available online.

In more detail, we address the data-to-text generation task using two settings: LLM prompting and fine-tuning. However, fine-tuning LLMs for specific tasks remains challenging, often constrained by computational resources. To mitigate this, we use a Parameter Efficient Fine-Tuning (PEFT) technique to substantially reduce the number of parameters participating in training, making the fine-tuning process far more computationally efficient while maintaining model performance. In both explored settings, we use an external Machine Translation (MT) system to translate our English-generated texts into Chinese, German, Russian, Spanish, Korean, Hindi, Swahili, and Arabic.

The paper is structured as follows. Section 2 describes data and task, and Section 3 presents the general approach, prompt types, testing phase and the specific systems we fully evaluated. Experimental set-up and results are outlined in Section 4, and Section 5 provides conclusions.

All the code and generated texts are available on GitHub.[1]

2 Data and Task

The Data-to-Text task converts input data, specifically RDF triples representing *subject* | *predicate* | *object* combinations, into coherent and contextually appropriate text that accurately conveys all and only the information present in the input triples.

The GEM 2024 shared task provides datasets for two subtasks: (i) WebNLG-based, utilising the official WebNLG (Castro Ferreira et al., 2020) test set, and (ii) Wikidata-based, using newly obtained triples from Wikidata. Each subtask includes three parallel datasets: Factual, Counterfactual, and Fictional. The Factual dataset consists of triples found in WebNLG or Wikidata. The Counterfactual dataset switches entities based on their class,

[1] https://github.com/michelalorandi/DCU-NLG-PBN-GEM24

Proceedings of the 17th International Natural Language Generation Conference: Generation Challenges, pages 76–83
September 23 –27, 2024.

creating hypothetical scenarios. Finally, the Fictional dataset replaces original entities with those created via LLM prompting.

For all datasets, only the test set is provided, containing the input triples with predicates in English. No training data is available, and reference texts are not provided. However, for the WebNLG-based Factual dataset, references can be extracted from the original WebNLG English dataset, allowing for some level of automatic evaluation.

3 Systems

We consider two settings to create our systems using pretrained LLMs (Figure 1): (i) generate text in English using out-of-the-box LLMs with prompting, (ii) generate text in English using a fine-tuned LLM. In the first setting, we employ pretrained LLMs without additional training and use various prompting strategies to guide the model in generating text based on the input RDF triples. In the second setting, we fine-tune pretrained LLMs using Low-Rank Adaptation (LoRA). Regardless of the generation method, the generated English text is then translated into Chinese, German, Russian, Spanish, Korean, Hindi, Swahili, and Arabic using a Machine Translation system.

3.1 Prompt types

In our experiments, we used the same prompts proposed by Lorandi and Belz (2023): Zero-shot minimal instruction and Few-shot in-context learning. **Zero-shot minimal instruction** consists of a simple and brief description of the task followed by the input. The prompt does not include any detail or example of the task. **Few-shot in-context learning** contains the same brief task description but adds a list of examples showing both input and target output.

We explored four variations of Few-shot in-context learning, each differing in how examples were selected, based on the idea that choosing examples similar to the input triples would improve the model's performance:

1. *Fixed examples*: The list of examples is fixed for every sample in the dataset.

2. *Dynamic examples based on triple set length*: Examples are randomly selected from the list where the triple set length matches the input triple set length.

3. *Dynamic examples based on properties*: Examples are randomly selected from those that share at least one property with the input; if no such examples exist, a random selection from all examples is performed.

4. *Dynamic examples based on triple set length and properties*: Examples are first filtered by matching triple set length, then randomly selected from those that share at least one property with the input; if no such examples exist, a random selection from the length-matched examples is performed.

3.2 Testing and model selection

We conducted a testing phase using the entire English validation set of WebNLG 2020 to evaluate our settings. We tested two instructed-tuned LLMs for prompting and four LLMs for fine-tuning, resulting in the following experimental grids:

1. {Mistral 7B Instruct, Falcon 40B Instruct} x {Zero Shot, Few Shot, Few Shot with dynamic examples based on triple set length, Few Shot with dynamic examples based on predicates, Few Shot with dynamic examples based on triple set length and predicates} x {English}

2. {Mistral 7B, Mistral 7B Instruct, Falcon 40B, Falcon 40B Instruct} x {WebNLG 2020 (English)} x {LoRA} x {English}

Prompting. We tested all the prompts described in Section 3.1 using Mistral 7B Instruct[2] and Falcon 40B Instruct.[3] For the dynamic selection of examples, we created a pool of possible examples from the train set and translated them into all languages using No Language Left Behind (NLLB) (Costa-jussà et al., 2022). All prompts were tested on the complete validation set of WebNLG 2020. The full text of the used prompts is shown in Appendix A.

Model Fine-Tuning. We PEFT-tuned four different LLMs: Mistral 7B (Jiang et al., 2023), Mistral 7B Instruct, Falcon 40B (Almazrouei et al., 2023), and Falcon 40B Instruct. We used LoRA (Hu et al., 2021) as the PEFT technique to fine-tune the selected models using the training and validation sets

[2] https://huggingface.co/mistralai/Mistral-7B-Instruct-v0.2

[3] https://huggingface.co/tiiuae/falcon-40b-instruct

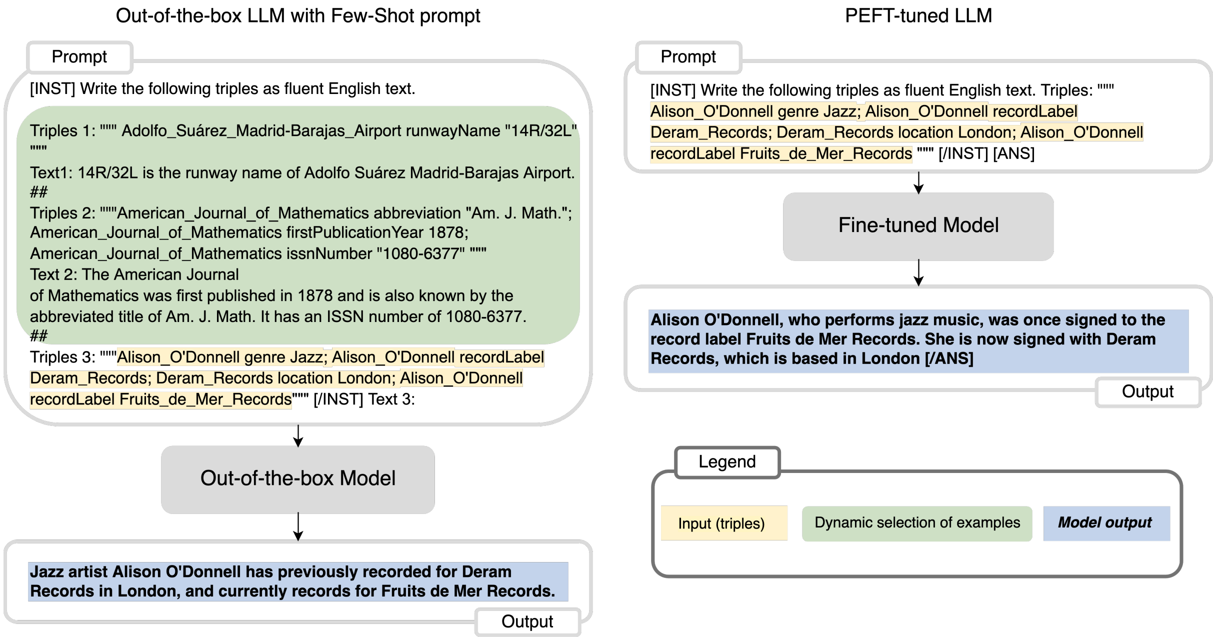

Figure 1: The two systems used in the final evaluation with input and output structure examples. Given Input (triples) highlighted in yellow, model output in blue. The few-shot in-context prompt also incorporates examples (highlighted green).

Model	Setting	BLEU↑	ChrF++↑	TER↓
Mistral 7B	Fine-tuning	**62.878**	**0.75**	**0.33**
Mistral 7B Instruct	Fine-tuning	55.1306	0.71	0.45
	Zero-shot	23.2855	0.58	0.82
	Few-shot fixed	36.8946	0.65	0.61
	Few-shot dynamic, length	36.3098	0.65	0.61
	Few-shot dynamic, properties	38.8017	0.66	0.57
	Few-shot dynamic, length and properties	*40.1638*	*0.67*	*0.55*
Falcon 40B	Fine-tuning	46.0399	0.5	0.55
Falcon 40B Instruct	Fine-tuning	46.0189	0.68	0.48
	Zero-shot	22.0014	0.24	0.82
	Few-shot fixed	25.9916	0.42	0.75
	Few-shot dynamic, length	18.5744	0.21	0.84
	Few-shot dynamic, properties	16.4993	0.17	0.89
	Few-shot dynamic, length and properties	22.2892	0.22	0.81

Table 1: Preliminary automatic evaluation results of our testing phase on the validation set of WebNLG 2020 in English. Best overall system in bold, best prompting system in italics.

of WebNLG 2020. For fine-tuning, we construe the task as an instruction-based task where special tokens delimit the task description, input, and output. The special tokens are designed to train the model to accurately identify the answer, ensuring it includes all and only the information contained in the input, thereby reducing hallucinations and omissions. See Section 4 for more details.

We performed post-processing based on the validation set results in both settings, as follows. We removed special tokens for the start of the sentence, end of the sentence, and padding. The answer was considered to be the text between the special answer tokens in the case of fine-tuning, and the text up to the first occurrence of the character sequence *Triples* or *(Note:* in the case of prompting. We further removed [and] characters and replaced \n with a space.

Table 1 shows the preliminary results from the testing phase. Mistral 7B consistently outperformed alternatives by substantial margins. Furthermore, within the prompting results, Mistral 7B Instruct with Few-Shot prompts using dynamic examples selected based on length and predicates outperformed all other prompting techniques. We selected these configurations as our final systems for submission based on these preliminary results.

Model	Setting	BLEU↑	BLEU NLTK↑	METEOR↑	ChrF++↑	BERT↑ P	R	F1
Mistral 7B Instruct	Fine-tuning	52.26	0.516	0.41	0.679	0.958	0.955	0.956
	Few-shot dynamic	40.12	0.395	0.401	0.655	0.946	0.954	0.949
GPT-3.5 (175B)	Few-shot fixed	**52.74**	**0.519**	**0.417**	**0.69**	**0.959**	**0.958**	**0.958**

Table 2: Automatic evaluation results comparison between our system and Lorandi and Belz (2023) best system (GPT-3.5) on the test set of WebNLG 2020 in English. Best overall system in bold. Few-shot dynamic = Few-shot prompt with dynamic selection of examples based on length and predicates.

System	BLEU↑	METEOR↑	ChrF++↑	BERT-F1↑
DCU-ADAPT-modPB	49.8	0.400	0.655	0.955
DCU-NLG-PBN (our)	**52.26**	**0.410**	0.679	**0.956**
DCU-NLG-Small	51.43	0.395	0.662	0.954
DipInfo-UniTo	51.36	**0.410**	**0.681**	0.955
OSU CompLing	43.09	0.389	0.65	0.950
RDFpyrealb	42.38	0.390	0.642	0.946
SaarLST	39.86	0.400	0.655	0.947

Table 3: Automatic evaluation results on the English test set of WebNLG 2020, comparing the performance of participating systems in the GEM 2024 shared task. Best overall system in bold.

3.3 Prompts and models used in final systems

Based on the results of our testing phase, we evaluated the following system variants as our final systems:

1. {Mistral 7B Instruct} x {Few Shot with dynamic examples based on triple set length and predicates} x {Google Translate} x {English, Chinese, German, Russian, Spanish, Korean, Hindi, Swahili, Arabic}

2. {Mistral 7B} x {WebNLG 2020 (English)} x {LoRA} x {Google Translate} x {English, Chinese, German, Russian, Spanish, Korean, Hindi, Swahili, Arabic}

Both systems incorporate the post-processing steps described in Section 3.2. We use Google Translate to translate English-generated texts into Chinese, German, Russian, Spanish, Korean, Hindi, Swahili, and Arabic.

4 Experimental Set-up and Results

We executed our experiments using the transformer library[4] of HuggingFace and the paid-for Google Translate API[5] in late March/early April 2024. The systems are tested using the six datasets described in Section 2. All generated texts are post-processed as described in Section 3.2. All systems are executed on a Nvidia A100 GPU with 80GB RAM.

Prompting. We set mistralai/Mistral-7B-Instruct-v0.2 parameters to *max seq length*=512, *seed*=6787, and *use 4bit*=True.

Model Fine-tuning. We use the PEFT library[6] of HuggingFace to create and load LoRA modules. We set mistralai/Mistral-7B-v0.1 parameters to *max steps*=10000, *learning rate*=2e-4, *max grad norm*=0.3, *weight decay*=0.001, *lora alpha*=16, *lora dropout*=0.1, *lora r*=64, *max seq length*=512, *seed*=6787, *use 4bit*=True, and *warmup ratio*=0.3. We use the checkpoint at step 6000 at inference time as it has the lowest loss based on the validation set. WebNLG 2020 train set is used for the model fine-tuning. The fine-tuning is defined as an instruction-based task where the task description and input are delimited by special instruction tokens ([INST] and [/INST]), and the output is delimited by special answer tokens ([ANS] and [/ANS]).

Following the WebNLG 2023 evaluation setup (Cripwell et al., 2023), we perform an automatic evaluation on the WebNLG-based Factual dataset in English computing BLEU (Papineni et al., 2002), ChrF++ (Popović, 2017), METEOR (Banerjee and Lavie, 2005), and BERTScore (Zhang et al.). We compare our two systems against the best system proposed by Lorandi and Belz (2023), i.e. GPT-3.5 using Few-Shot prompt with fixed examples.

An additional human evaluation will be performed by the organisers of the shared task and

[4]https://huggingface.co/docs/transformers/index

[5]https://cloud.google.com/translate

[6]https://huggingface.co/docs/peft/index

at the time of writing the results are not available yet. Refer to the shared task report for more details.

Table 2 shows the results of the automatic evaluation in English on the WebNLG-based Factual dataset, for which references are available. Our fine-tuned model outperforms the prompting-based Mistral 7B Instruct by clear margins. Scores for GPT-3.5 are higher than for fine-tuned Mistral 7B Instruct by tiny margins in all cases. However, the latter achieves these very close results while utilising a substantially smaller model size (25x). This significant reduction in model size translates to lower computational costs, decreased memory usage, and faster processing times, making the fine-tuned Mistral 7B a more resource-efficient option.

Table 3 shows the automatic evaluation results in English on the WebNLG-based Factual dataset comparing all participating systems in the GEM 2024 shared task. Our fine-tuned system (DCU-NLG-PBN) shows strong performance, achieving the highest scores in both BLEU and BERT-F1. DipInfo-UniTo system, while slightly lower in BLEU, leads in ChrF++ and performs competitively in METEOR, alongside our system. These results, however, represent partial evaluations on the WebNLG-based Factual dataset using all available references. More insights on the performance of the systems will emerge from the human evaluation results. For additional automatic evaluation results, refer to the shared task report (Mille et al., 2024).

5 Conclusion

We explored the effectiveness of pretrained LLMs for data-to-text generation focusing on two settings: LLM prompting and LLM fine-tuning with LoRA. We first conducted a testing phase comparing the performance of Mistral 7B and Falcon 40B models using various prompting strategies and fine-tuning techniques, evaluated on the WebNLG 2020 validation set. The results demonstrated that fine-tuning with LoRA substantially enhances the performance of the Mistral 7B model. This model outperformed all other tested systems, including Falcon 40B. Among the prompting strategies, the few-shot in-context learning with dynamic examples based on the triple set length and predicates achieved the best results, indicating the importance of contextually relevant example selection.

We submitted the two system settings, LLM prompting + MT and LLM fine-tuning with LoRA + MT, using Mistral 7B to the GEM 2024 shared task in English, Chinese, German, Russian, Spanish, Korean, Hindi, Swahili, and Arabic.

Our findings highlight the potential of LoRA for efficient fine-tuning of LLMs, offering a competitive performance close to state-of-the-art models like GPT-3.5, but with substantially smaller model sizes and reduced resource requirements. The success of dynamic example selection in prompting also underscores the need for tailored approaches to optimize model performance.

Acknowledgments

Michela Lorandi's work was conducted with the financial support of the Science Foundation Ireland Centre for Research Training in Digitally-Enhanced Reality (d-real) under Grant No. 18/CRT/6224. Both authors benefit from being members of the ADAPT SFI Research Centre at Dublin City University, funded by the Science Foundation Ireland under Grant Agreement No. 13/RC/2106_P2. For the purpose of Open Access, the author has applied a CC BY public copyright licence to any Author Accepted Manuscript version arising from this submission.

References

Ebtesam Almazrouei, Hamza Alobeidli, Abdulaziz Alshamsi, Alessandro Cappelli, Ruxandra Cojocaru, Merouane Debbah, Etienne Goffinet, Daniel Heslow, Julien Launay, Quentin Malartic, Badreddine Noune, Baptiste Pannier, and Guilherme Penedo. 2023. Falcon-40B: an open large language model with state-of-the-art performance.

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.

Marta R Costa-jussà, James Cross, Onur Çelebi, Maha Elbayad, Kenneth Heafield, Kevin Heffernan, Elahe Kalbassi, Janice Lam, Daniel Licht, Jean Maillard, et al. 2022. No language left behind: Scaling human-centered machine translation. *arXiv preprint arXiv:2207.04672*.

Liam Cripwell, , Anya Belz, Claudia Borg, Claire Gardent, Albert Gatt, John Judge, Michela Lorandi, Anna Nikiforoskaya, William Soto-Martinez, and Craig Thomson. 2023. The 2023 webnlg shared task on low resource languages overview and evaluation results (webnlg 2023). In *Proceedings of the Workshop on Multimodal, Multilingual Natural Language Generation and Multilingual WebNLG Challenge*, Prague, Czech Republic.

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

Albert Q Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, et al. 2023. Mistral 7b. *arXiv preprint arXiv:2310.06825*.

Michela Lorandi and Anya Belz. 2023. Data-to-text generation for severely under-resourced languages with GPT-3.5: A bit of help needed from Google Translate (WebNLG 2023). In *Proceedings of the Workshop on Multimodal, Multilingual Natural Language Generation and Multilingual WebNLG Challenge (MM-NLG 2023)*, pages 80–86, Prague, Czech Republic. 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.

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.

Maja Popović. 2017. chrF++: words helping character n-grams. In *Proceedings of the Second Conference on Machine Translation*, pages 612–618, Copenhagen, Denmark. Association for Computational Linguistics.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q Weinberger, and Yoav Artzi. Bertscore: Evaluating text generation with bert. In *International Conference on Learning Representations*.

A Prompts

Table 4 shows the input template of our fine-tuned Mistral 7B Instruct. Tables 5 and 6 shows the Zero-shot prompts used with Mistral 7B Instruct and Falcon 40B Instruct, while Tables 7 and 8 shows the Few-shot prompts used with Mistral 7B Instruct and Falcon 40B Instruct.

Input template fine-tuned Mistral 7B	
Template:	[INST] Write the following triples as fluent English text. Triples: """ {set of triples in the format *subject predicate object* and each triple separated by ;} """ [/INST] [ANS] {MODEL} [/ANS]
Example:	[INST] Write the following triples as fluent English text. Triples: """ Alison_O'Donnell genre Jazz; Alison_O'Donnell recordLabel Deram_Records; Deram_Records location London; Alison_O'Donnell recordLabel Fruits_de_Mer_Records """ [/INST] [ANS]
Model output:	*Alison O'Donnell, who performs jazz music, was once signed to the record label Fruits de Mer Records. She is now signed with Deram Records, which is based in London [/ANS]*

Table 4: Input template for our fine-tuned Mistral 7B. **Top** Template of the input. **Bottom** Complete example in English.

Zero-Shot prompt Mistral 7B Instruct	
Template:	[INST] Write the following triples as fluent English text. Triples: """ {set of triples in the format *subject predicate object* and each triple separated by ;} """ [/INST] Text: {MODEL}
Example:	[INST] Write the following triples as fluent English text. Triples: """ Alison_O'Donnell genre Jazz; Alison_O'Donnell recordLabel Deram_Records; Deram_Records location London; Alison_O'Donnell recordLabel Fruits_de_Mer_Records """ [/INST] Text:
Model output:	*Alison O'Donnell is a jazz artist. She is signed under the record label Deram Records. Deram Records is based in London. Alternatively, Alison O'Donnell has also recorded under the record label Fruits de Mer Records.*

Table 5: Zero-shot prompt for Mistral 7B Instruct. **Top** Template of the input. **Bottom** Complete example in English.

Zero-Shot prompt Falcon 40B Instruct	
Template:	»QUESTION« Write the following triples as fluent English text. Triples: """ {set of triples in the format *subject predicate object* and each triple separated by ;} """ »ANSWER« Text: {MODEL}

Table 6: Zero-shot prompt for Falcon 40B Instruct.

Few-Shot prompt Mistral 7B Instruct	
Template:	[INST] Write the following triples as fluent English text. Triple 1: """ {set of triples in the format *subject predicate object* and each triple separated by ;} """ Text 1: {verbalisation of Triple 1} ## Triple 2: """ {set of triples in the format *subject predicate object* and each triple separated by ;} """ Text 2: {verbalisation of Triple 2} ## Triple 3: """ {set of triples in the format *subject predicate object* and each triple separated by ;} """ [/INST] Text 3: {MODEL}
Fixed examples:	Triple set 1: Adolfo_Suárez_Madrid–Barajas_Airport runwayName "14R/32L" Text 1: 14R/32L is the runway name of Adolfo Suárez Madrid-Barajas Airport. Triple set 2: American_Journal_of_Mathematics abbreviation "Am. J. Math."; American_Journal_of_Mathematics firstPublicationYear 1878; American_Journal_of_Mathematics issnNumber "1080-6377" Text 2: The American Journal of Mathematics was first published in 1878 and is also known by the abbreviated title of Am. J. Math. It has an ISSN number of 1080-6377.
Example Prompt:	[INST] Write the following triples as fluent English text. Triple 1: """ Adolfo_Suárez_Madrid–Barajas_Airport runwayName "14R/32L" """ Text 1: 14R/32L is the runway name of Adolfo Suárez Madrid-Barajas Airport. ## Triple 2: """ American_Journal_of_Mathematics abbreviation "Am. J. Math."; American_Journal_of_Mathematics firstPublicationYear 1878; American_Journal_of_Mathematics issnNumber "1080-6377" """ Text 2: The American Journal of Mathematics was first published in 1878 and is also known by the abbreviated title of Am. J. Math. It has an ISSN number of 1080-6377. ## Triple 3: """ Alison_O'Donnell genre Jazz; Alison_O'Donnell recordLabel Deram_Records; Deram_Records location London; Alison_O'Donnell recordLabel Fruits_de_Mer_Records """ [/INST] Text 3:
Model output:	*Jazz artist Alison O'Donnell has previously recorded for Deram Records in London, and currently records for Fruits de Mer Records.*

Table 7: Few-Shot prompt for Mistral 7B Instruct. **Top** Template of the prompt. **Middle** Fixed examples used during testing. **Bottom** Complete example in English.

Few-Shot prompt Falcon 40B Instruct	
Template:	»QUESTION« Write the following triples as fluent English text. Triple 1: """ {set of triples in the format *subject predicate object* and each triple separated by ;} """ Text 1: {verbalisation of Triple 1} ## Triple 2: """ {set of triples in the format *subject predicate object* and each triple separated by ;} """ Text 2: {verbalisation of Triple 2} ## Triple 3: """ {set of triples in the format *subject predicate object* and each triple separated by ;} """ »ANSWER« Text 3: {MODEL}
Fixed examples:	Triple set 1: Adolfo_Suárez_Madrid–Barajas_Airport runwayName "14R/32L" Text 1: 14R/32L is the runway name of Adolfo Suárez Madrid-Barajas Airport. Triple set 2: American_Journal_of_Mathematics abbreviation "Am. J. Math."; American_Journal_of_Mathematics firstPublicationYear 1878; American_Journal_of_Mathematics issnNumber "1080-6377" Text 2: The American Journal of Mathematics was first published in 1878 and is also known by the abbreviated title of Am. J. Math. It has an ISSN number of 1080-6377.

Table 8: Few-Shot prompt for Falcon 40B Instruct. **Top** Template of the prompt. **Bottom** Fixed examples used during testing.