Generative AI - Project Report
M2 TAL

Binesh Arakkal Remesh, Zahra Sharifi, Elise Soenen

Université de Lorraine - IDMC

1 Introduction

This experiment aimed to perform a meaning-representation to text generation. To complete this task, we used the pre-trained model T5-small developed at Google, as well as the E2E NLG dataset. T5-small is a sequence-to-sequence model and shows better performance in text-to-text generation, making it the best choice for our task.

Dataset The publicly available dataset E2E NLG (Novikova et al., 2017) serves as an English evaluation benchmark for models that convert structured data into natural language text within the restaurant context. It works with sets containing between 2 and 9 attribute-value pairs of the form *attribute*[*value*]. The GEM implementation utilizes a refined version of this dataset (Dušek et al., 2019), which removes instances containing fabricated information and incomplete representations of the input attributes. It can be retrieved from this website: `https://gem-benchmark.com/data_cards/e2e_nlg`.

Model T5-small ("Text-to-Text Transfer Transformer")(Raffel et al., 2020) is developed by Google and has an encoder-decoder architecture with 60 million parameters. It uses text-to-text format, where the input and output are always text strings. T5-small was pre-trained on the Colossal Clean Crawled Corpus (C4) (Raffel et al., 2020), which was developed and released in the context of the same research paper as T5. The model supports several languages, such as English, French, Romanian, and German.

It was trained on various NLP tasks such as Sentence Acceptability, Paraphrasing/Sentence Similarity, Natural Language Inference, Coreference Resolution, sentence completion, Word Sense Disambiguation, Question Answering, and more. As an example, Pasricha et al., 2020 used T5-small for RDF-to-Text generation task. The comparatively small size of the model allows us not to load too much of the GPU memory and makes it efficient. These reasons make the model apt for the chosen task. More documentation can be retrieved from this link: `https://huggingface.co/google-t5/t5-small`.

2 Method

Our approach was to test the T5-small on the pre-processed test data from E2E NLG corpus and evaluate the performance. Then we fine-tuned the T5-small on

the pre-processed train and validation sets of E2E NLG dataset, and compared the results with the generated output before and after the fine-tuning. For pre-processing, a linearization function was defined which changes the attribute–value pairs into sequence-type data, like explicits with this pattern example:

$$attribute[value] \text{ to } attribute = value$$

2.1 Fine-tuning

Once the data were in a processable shape by removing unnecessary columns and linearization, a tokenization function was defined and mapped with the train and validation dataset. The training was done using sequence-to-sequence trainer with sequence-to-sequence argument. The parameters of the argument are given below: number of training epochs of 3, training batch size of 16, evaluation batch size of 32, the gradient accumulation steps of 2. We used an evaluation strategy 'steps', with an eval_steps equals to 500 to evaluate the model during training steps to avoid overfitting.

2.2 Evaluation Metrics

We used four metrics to evaluate the performance of the model, before and after fine-tuning. 'BLEU', predominantly used to evaluate the automatic translations performed by a machine translator, uses the n-grams overlaps between the generated text and the reference text. Some previous studies on graph-to-text generation use BLEU for evaluating the model for example (Yuan and Färber, 2023).

The second metric used for the evaluation is 'METEOR' (Banerjee and Lavie, 2005), which checks the alignment between the generated output text and the reference texts. 'ROUGE' score and chrF++ (Lin, 2004; Popović, 2017) were the other metrics used in this project. These scores measure overlaps of n-grams, longest common subsequences (LCS), word sequences between generated and reference texts, and the quality of machine-generated texts, respectively.

The score 60.04 for chrf++ shows that the generated outputs have moderate - high overlap between the reference sentence in the E2E NLG corpus.
All the code and output dataset are available in this repository: `https://github.com/Elise-S/GenAI-Project-2025`.

3 Results

The performance of T5-small before fine-tuning on E2E NLG corpus was unsatisfactory. The model was outputting the input sentence itself, which demands fine-tuning the model on the train and validation set of E2E NLG corpus. After fine-tuning the Model on E2E NLG corpus, the model showed a significant amount of change on the performance.

Table 1 shows comparison between the raw model outputs against the fine-tuned model outputs. We can see that all the evaluation metrics, without exception, possess a significant increase after the fine-tuning process. It is clear that the fine-tuned

Evaluation Metric	Score	
	Before Fine-tuning	**After Fine-tuning**
BLEU	0.0626	**0.4344**
METEOR	0.3681	**0.6560**
ROUGE-1	0.5071	**0.7304**
ROUGE-2	0.2388	**0.4741**
ROUGE-L	0.3758	**0.5638**
chrF++	41.60	**60.04**

Table 1: Evaluation results of the T5-small model on the E2E NLG dataset before and after fine-tuning

T5 model performed better. For the BLUE score, we can notice that the generated output of the fine-tuned model is similar to the reference text, and this applies to all the evaluation metrics.

4 Conclusion

In summary, fine-tuning T5-small on the E2E NLG corpus was significant for the task of generating text from meaning-representation, as expected. Meaning representation was well handled by the T5-small model. Before fine-tuning, the model failed to give any qualitative outputs. After fine-tuning the generated answers not only seem grammatical and fluent but also make sense. Some examples are given in the box 4 available in the Annexe section.

References

Banerjee, S. and Lavie, A. (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.

Dušek, O., Howcroft, D. M., and Rieser, V. (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.

Lin, C.-Y. (2004). ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain. Association for Computational Linguistics.

Novikova, J., Dušek, O., and Rieser, V. (2017). The e2e dataset: New challenges for end-to-end generation. *arXiv preprint arXiv:1706.09254*.

Pasricha, N., Arcan, M., and Buitelaar, P. (2020). Nuig-dsi at the webnlg+ challenge: Leveraging transfer learning for rdf-to-text generation. In *Proceedings of the 3rd International Workshop on Natural Language Generation from the Semantic Web (WebNLG+)*, pages 137–143.

Popović, M. (2017). chrF++: words helping character n-grams. In Bojar, O., Buck, C., Chatterjee, R., Federmann, C., Graham, Y., Haddow, B., Huck, M., Yepes, A. J., Koehn, P., and Kreutzer, J., editors, *Proceedings of the Second Conference on Machine Translation*, pages 612–618, Copenhagen, Denmark. Association for Computational Linguistics.

Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., Zhou, Y., Li, W., and Liu, P. J. (2020). Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140):1–67.

Yuan, S. and Färber, M. (2023). Evaluating generative models for graph-to-text generation. *arXiv preprint arXiv:2307.14712*.

ANNEXE

A. Output Examples (Before finetuning the model)

Example Output 1: **data['train'][555]**

Graph: name[The Mill], eatType[pub], priceRange[cheap], customer rating[high], area[riverside], familyFriendly[no], near[Café Rouge]

Input: translate MR to text: name = The Mill ; eatType = pub ; priceRange = cheap ; customer rating = high ; area = riverside ; familyFriendly = no ; near = Café Rouge

References: 1. In the riverside, near Café Rouge, there is The Mill, a cheap highly rated non-kids friendly pub. 2. The Mill is a cheap highly rated non-kids friendly pub, located in the riverside, near Café Rouge.

Model Output before Fine-tuning: 'MR to text: name = The Mill ; eatType = pub ; priceRange = cheap ; customer rating = high ; area = riverside ; familyFriendly = no ; near = Café Rouge'. Example of the generated output

B. Output Examples (Fined-tuned model)

Example Output 1: **data['train'][555]**

Graph: name[The Mill], eatType[pub], priceRange[cheap], customer rating[high], area[riverside], familyFriendly[no], near[Café Rouge]

Input: translate MR to text: name = The Mill ; eatType = pub ; priceRange = cheap ; customer rating = high ; area = riverside ; familyFriendly = no ; near = Café Rouge

References: 1. In the riverside, near Café Rouge, there is The Mill, a cheap highly rated non-kids friendly pub. 2. The Mill is a cheap highly rated non-kids friendly pub, located in the riverside, near Café Rouge.

Model Output: The Mill is a cheap pub near Café Rouge in the riverside area. It is not family-friendly and has a high customer rating. Example of the generated output

Example output 2: **data['train'][26]**

Graph: name[Giraffe], eatType[pub], near[The Bakers]', 'input': 'translate MR to text: name = Giraffe ; eatType = pub ; near = The Bakers

Input: translate MR to text: name = Giraffe ; eatType = pub ; near = The Bakers.

References: 1. A pub named Giraffe is near The Bakers.
2. Giraffe is a pub found near The Bakers.

Model Output: Giraffe is a pub near The Bakers