LDK 2023

Improving Graph-to-Text Generation Using Cycle Training

Fina Polat
University of Amsterdam
f.yilmazpolat@uva.nl

Ilaria Tiddi
Vrije Universiteit Amsterdam
i.tiddi@vu.nl

Paul Groth
University of Amsterdam
p.t.groth@uva.nl

Piek Vossen
Vrije Universiteit Amsterdam
p.t.j.m.vossen@vu.nl

Abstract

Natural Language Generation (NLG) from graph structured data is an important step for a number of tasks, including e.g. generating explanations, automated reporting, and conversational interfaces. Large generative language models are currently the state of the art for open ended NLG for graph data. However, these models can produce erroneous text (termed hallucinations). In this paper, we investigate the application of *cycle training* in order to reduce these errors. Cycle training involves alternating the generation of text from an input graph with the extraction of a knowledge graph where the model should ensure consistency between the extracted graph and the input graph. Our results show that cycle training improves performance on evaluation metrics (e.g., METEOR, DAE) that consider syntactic and semantic relations, and more in generally, that cycle training is useful to reduce erroneous output when generating text from graphs.

1 Introduction

Graph-to-Text generation (G2T) is a subtask of open-ended Natural Language Generation (NLG) that aims to create fluent natural language text describing an input graph, and is part of common NLG benchmarks (Gehrmann et al., 2021). G2T conversion is particularly of interest for open-ended generation tasks such as dialogue generation and generative question answering (Ribeiro et al., 2021; Trisedya and et al., 2019). Large generative language models are currently the state of the art for open ended NLG from graph data (Gehrmann et al., 2021). A major problem faced by these models is the output of non-sensical or unfaithful content to the provided input. This phenomenon is known as hallucination (Ji et al., 2022).

Figure 1 displays an example of Graph-to-Text conversion. The NLG model, a large language model (T5-small, Raffel et al. (2020)) is finetuned with a widely used benchmark corpus (WebNLG,

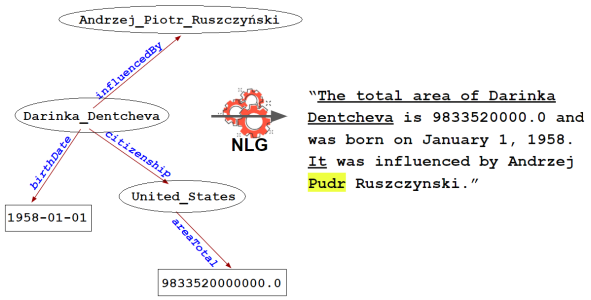

Figure 1: Graph-to-Text generation example with a hallucinatory verbalization.

Zhou and Lampouras (2020)), is asked to convert a graph taken from WebNLG. The output contains several errors. For example, *Darinka Dentcheva* is mentioned, as if she were a location, and attributed a total area. The generation continues with a proper verbalization of *birthDate*, but then again the model fails by referring *Darinka Dentcheva* with the pronoun *it*. Another mistake is the generation of an incorrect name. *Andrzej **Piotr** Ruszczynski* becomes *Andrzej **Pudr** Ruszczynski*.

Hallucinations are divided into two categories (Ji et al., 2022): intrinsic and extrinsic. In Figure 1, the intrinsic hallucinations are underlined, and the extrinsic hallucination is highlighted. Intrinsic hallucinations are the generation of output that contradicts the input graph, does not make sense, or contains some sort of commonsense violation. Extrinsic hallucinations are generations that cannot be verified by the source. Thus, the output can neither be supported nor contradicted by the input graph.

In this paper, we aim at addressing these problems by employing cycle training. Cycle training makes use of inverse tasks to add the model with additional signals. Here, the inverse task of G2T is Text-to-Graph (T2G) conversion where structures in the form of knowledge graphs are extracted from the text. In particular, we propose to use the T2G component of the cycle training to detect hallucinatory information in the generation by comparing

the extracted triples with the input triples. Additionally, combining G2T and T2G conversions is expected to improve the quality of the generated text and faithfulness of an NLG system because we hypothesize that cycle training would teach the NLG model to remain faithful to the input graph with the support of cycle consistency. Therefore, combining these two tasks is thought to improve the quality of the verbalization and reduce the hallucinatory generation. Our full code is available online.[1]

The contributions of this paper are as follows:

1. An approach that employs cycle training to improve NLG faithfulness by reducing hallucinatory generation. Specifically, the approach introduces a T2G component to detect entity and relation mentions that are not part of the input graph.

2. A performance evaluation of this approach using three traditional lexical overlap metrics and two entailment evaluation methods used in the hallucination literature and show that the metrics with linguistic foundations (e.g. METEOR(+6%), DAE(+5%)) show significant improvement with cycle training.

2 Related Work

In recent years, there has been a paradigm shift in NLG. The shift stems from improvements in deep contextual language modeling and transfer learning (Ji and et al., 2020). NLG systems typically prioritize being coherent and discourse-related, disregarding control over generated content and its qualities such as faithfulness, factuality, freshness, and correctness. However, having control over the output is a major factor in NLG applications within industry (Leng and et al., 2020). Since cycle training reinforces the faithfulness of the NLG model and has the potential to detect extra information that is not part of the input, we relate our work to this controlability literature.

The state-of-the-art G2T generation results come from large generative models, but it is well known that these models are prone to hallucination. It is important to notice that all NLG tasks suffer from the hallucinatory text generation, and a control mechanism to solve this problem has not been found yet (Ji et al., 2022).

[1] https://github.com/cltl-students/fina_polat_nlg_with_transformers.

Leveraging the fact that two functions are inverse of each other has been widely used in a variety of tasks in computer vision and machine translation (Godard et al., 2017; Sennrich et al., 2016). In the context of G2T, cycle training is used to address parallel data scarcity. Parallel graph-text data collection is difficult and costly. Therefore supervised approaches to both G2T and T2G conversions suffer from a shortage of domain-specific parallel graph-text data. Guo et al. (2020) and Schmitt et al. (2020) propose cycle training approach as an unsupervised learning solution when there is no or limited parallel data.

Guo et al. (2020) employ high-performing Named Entity Recognition (NER) tools such as Stanza (Qi and et al., 2020) to extract the entities and then build graphs with these automatically extracted entities. They train a G2T model called CycleGT using these automatically built graphs as the input graph in a cycle training regime. They test their unsupervised approach on parallel graph-text datasets such as WebNLG to compare their results with supervised approaches. We build on this work but instead of focusing on addressing the problem of data scarcity, we focus on the problem of hallucinations.

3 Approach

Our approach uses supervised cycle training with the objective of cycle consistency. Specifically, we employ CycleGT from Guo et al. (2020) and train it from scratch for five epochs. As our baseline, we use a pre-trained generative language model, the small version of T5, and finetune it for five epochs as well. For the training of CycleGT and the finetuning of the baseline T5, we use the WebNLG Dataset with the given train-test split. However, our approach is data and model agnostic and all components could be replaced with alternatives.

CycleGT is originally designed to address the parallel data scarcity and to be used as an unsupervised learning method when there is no or limited graph annotation. In the unsupervised setup, Guo et al. (2020) reduce the graph extraction task to relation prediction and rely on the Stanza NER module to extract the entities. Their results show that this approach works well to tackle parallel data scarcity. However, we are not interested in the unsupervised approach because we do not tackle the data scarcity problem, but instead we aim at less hallucinatory G2T generation.

As our objective is to improve the quality of the generated text by reducing/eliminating extrinsic hallucinations, supervision is essential for our case. We assume high-quality parallel graph-text data is given, and we rely on cycle consistency for improving generation quality, and T2G module for detecting extrinsic hallucinations. To the best of our knowledge, this is the first attempt to investigate cycle training in G2T for reducing/eliminating extrinsic hallucinations, reinforcing model faithfulness, and overall generation quality.

We compare the performance of CycleGT to the T5 baseline. All the experiments are run on a personal laptop. We now describe the data and models in more detail.

3.1 Data

WebNLG (Zhou and Lampouras, 2020) is a widely used G2T corpus that is created from DBpedia (Mendes and et al., 2011). DBpedia is a multilingual knowledge base that was built from various kinds of structured information contained in Wikipedia. This data is stored as RDF[2] triples, complies with Linked Data standards, and results in a high-quality dataset.[3]

3.2 Models

We choose T5 (Raffel et al., 2020) as the baseline pretrained language model, because it is state-of-the-art on the WebNLG dataset. Furthermore, T5 is a good representative sample of a generative large language model. We experiment with CycleGT because its G2T module is also based on T5 architecture that makes comparison easier. However, CycleGT does not exploit the pretrained language model but only utilize the architecture.

3.2.1 Baseline - T5

The "Text-to-Text Transfer Transformer" (or T5) is a unified framework that converts all text-based language problems into a text-to-text format (Raffel et al., 2020). The basic idea underlying the T5 model is to treat every textual task as a translation from input text to output text. In our case, the task consists in taking RDF triples as input, and producing a new text describing these triples as the output.

We finetune the small version of T5 model with the given train-test split of WebNLG for five epochs using Transformers library (Wolf and et al., 2020).

3.2.2 CycleGT

The G2T module of CycleGT transforms the graph to text. And, the T2G converts text to the graph by aligning each text with its back-translated version, and also each graph with its back-translated version. Since pretrained language models are shown to be effective on G2T conversions, Guo et al. (2020) use T5 (Raffel et al., 2020) architecture as the G2T component.

T2G produces a graph based on the given text. Guo et al. (2020) see relation extraction as the core problem in T2G conversion. In the supervised setup, T2G module of CycleGT directly uses the entities as they are given. Relations are predicted between every two pairs of entities with an LSTM-based Neural Network to form the edges in the graph. For our experiments, CycleGT is trained for five epochs in a supervised setup.

4 Evaluation

Considering the difficulty of quantifying hallucination, we use five different metrics for evaluation and divide them into two categories. The first category solely relies on lexical (n-gram) overlap while the second group is based on textual entailment.

4.1 Lexical Overlap Metrics

Lexical overlap metrics are widely used in NLG. The central idea behind these metrics is closeness. One of the simplest approaches is to leverage lexical features (n-grams) to calculate the similarity between the generation and the target text. We use BLEU (Papineni and et al., 2002), ROUGE (Lin, 2004), and METEOR (Banerjee and Lavie, 2005) as the lexical overlap metrics.

4.2 Entailment Metrics

Apart from well-established lexical overlap evaluation metrics, textual entailment models have been employed to evaluate the quality of automatically generated text. The entailment evaluation models are shaped around the idea that all information in the generated text should be entailed/inferred by the reference (gold) text.

For the evaluation of our NLG models, we employ two metrics that leverage entailment models: PARENT (Dhingra et al., 2019) and DAE (Goyal and Durrett, 2021).

[2]Resource Description Framework: https://www.w3.org/RDF/

[3]https://gitlab.com/shimorina/webnlg-dataset/-/tree/master/release_v3.0

Model	BLEU	ROUGE	METEOR	PARENT			DAE
				Precision	Recall	F1 score	
T5-small	19.6257	0.5668	0.4157	0.1910	0.0976	0.0939	0.2347
CycleGT	20.9327	0.5463	0.4740	0.1980	0.0894	0.0927	0.2829

Table 1: Graph-to-Text module evaluation scores.

4.2.1 PARENT

Lexical overlap metrics (BLEU, ROUGE, METEOR etc.) leverage the target text as the reference, and they do not take the input graph into account for the evaluation. However, it is common for a graph verbalization to have multiple plausible outputs from the same input.

Precision And Recall of Entailed N-grams from the Table, or PARENT, compares the generated text to the underlying graph as well as the reference text to improve evaluation. When computing precision, PARENT uses a union of the reference and the graph, to reward correct information missing from the reference. When computing recall, it uses an intersection of the reference and the graph, to ignore extra/incorrect information in the reference. The union and intersection are computed with the help of an entailment model to decide if an n-gram is entailed by the graph.

4.2.2 DAE

The DAE, or Dependency Arc Entailment, evaluation method is inspired by the downstream application of textual entailment models. Goyal and Durrett (2020) propose another formulation of the entailment that decomposes it at the level of dependency arcs. Rather than focusing on aggregate decisions, they instead ask whether the semantic relationship manifested by individual dependency arcs in the generated output is supported by the input. Arc entailment is a 2-class classification: entailed or not-entailed. This means that arcs that would be neutral or contradictory in the generic entailment formulation are considered non-entailed.

This approach views dependency arcs as semantic units that can be interpreted in isolation. Each arc is therefore judged independently based on whether the relation it implies is entailed by the reference sentence. A dependency arc in the generated sentence is assumed to be entailed by the reference if the semantic relationship between its head and child holds for the reference sentence. If the dependency relation does not hold for a head-child pair, then it is considered a factual error, and the mismatched head-child span can be marked as the hallucinatory generation.

4.3 Human Evaluation: Qualitative Analysis

Automatic evaluation metrics struggle to deal with semantic or syntactic variations. Therefore, we need human judgment even though it is costly. For qualitative analysis, we sample 100 instances from the test set, and one annotator performs the annotations following a two step annotation scheme. First, we annotate whether the generation contains any hallucination, a binary decision. If the generation is hallucinatory, we add the hallucination type, one of the following classes: intrinsic, extrinsic, or both.

5 Results and Discussions

Due to the limited compute resources, we choose smaller models, and train or finetune them for just five epochs. Therefore, the performance of our models could not reach to the range in other NLG experiments. However, we observe noticeable improvement in METEOR and DAE scores. We now detail the results of our experiments.

5.1 Automatic Evaluation Results

In Table 1, we report the results of the automatic evaluation metrics. ROUGE and METEOR scores are reported in terms of F1 score. For readability, the highest scores are underlined.

The CycleGT model trained in cycle consistency outperforms the finetuned T5 model in precision-oriented metrics: +1,3070 BLEU score and +0,0070 PARENT-precision. However, the finetuned T5 model takes the lead in terms of ROUGE (+0,0205) and PARENT-recall (+0,0082) scores. Precision and recall results of PARENT are consistent with BLEU and ROUGE. This is expected because BLEU is a precision-oriented score while ROUGE is recall oriented.

It is notable that CycleGT gets higher scores in terms of METEOR (+0,0583) and DAE (+0,0482). Compared to the precision-oriented scores, the difference in METEOR and DAE is more significant.

Both METEOR and DAE are built on evaluation models with a linguistic backup. METEOR, for instance, not only compares the text as a direct string match but also exploits synonymy. For a linguistically sound comparison, it uses the Porter Stemmer and WordNet as lexical database. Similarly, DAE is empowered by a dependency parsing framework.

METEOR and DAE are both empowered by linguistic backup, and they are designed to be able to measure the quality of a generation on higher levels, e.g. semantics. The shortcoming of these models is that the linguistic enhancements are also built on sub-modules, off-the-shelf tools, and automatically created datasets that are known to be prone to error propagation. Regardless of their flaws, METEOR and DAE are more advanced evaluation methods enhanced with linguistic backup compared to their alternatives. We also argue that the higher performance of CycleGT in terms of METEOR and DAE is indicative that these metrics are more suitable to automatically judge the quality of a generation.

5.2 Evaluation of the T2G Component

The evaluation of the T2G module of CycleGT is important due to three reasons. First, we expect CycleGT model to generate better and less hallucinatory (at least on the extrinsic side) text because it is trained in cycle consistency. The second reason is that we employ the T2G module of CycleGT to detect extrinsic (not part of the input, but made up by the NLG model) hallucinations in the generation. Therefore, it is supposed to be able to extract all the information in the generated text. Finally, both modules (G2T & T2G) are supposed to be equally strong for getting the maximum benefit from cycle training.

T2G	F1 Score overall	F1 Score partial	% of predictions
CycleGT	0.1407	0.7873	32%

Table 2: Evaluation scores of the Text-to-Graph module.

In Table 2, we report the evaluation results of the CycleGT T2G module. F1 scores are micro averaged. The T2G module displays recall deficiency. The overall performance of the graph extraction module is pretty poor (0.14 F1 score). The module usually fails to make at least one prediction per instance. The maximum number of predictions is 1662 (32%) out of 5150 test instances. This means that the model is unable to extract any triples from 68% of the test instances. However, it makes precise predictions when it does as indicated by the higher partial F1 score (0.78).

The poor performance of the T2G module of CycleGT reduces the robustness of cycle training. In order to enforce cycle consistency, a stronger T2G performance is necessary. Moreover, it is not possible to detect extrinsic hallucinations with this performance. Capturing extrinsic hallucinations would only be possible by a comparison between the input triples and the extracted triples. Therefore, it would be beneficial to aim at a better-performing triple extraction model to detect extrinsic hallucinations and reinforce cycle consistency.

5.3 Human Evaluation Results

Model	Only Intrinsic	Only Extrinsic	Both Int.&Ext
T5-small	11%	21%	20%
CycleGT	34%	18%	10%

Table 3: Qualitative Results.

Table 3 presents human evaluation results. This qualitative analysis confirms that CycleGT generates fewer extrinsic hallucinations. In our test sample, 18% of the CycleGT generations contain extrinsic hallucinations while the finetuned T5 model has 41%. Looking at the percentage of intrinsic hallucinations, the T5 model displays a better performance. On the one hand, we observe the generation of CycleGT mostly remains faithful to the input graph but contains wrong lexical associations (34%) with entities and their relations that occur as intrinsic hallucinations. On the other hand, we see that the finetuned T5 model makes more precise associations between entities and their relations but often makes up new entity names that were not part of the graph input (extrinsic hallucinations).

6 Conclusion

The use of generative models for NLG has led to improved performance, however, these models can still produce text with erroneous statements (i.e. hallucinations). In this paper, we show that combining G2T and T2G conversions in a cycle training setup helps such models improve the generated text conditioned on graph data. Automatic evaluation is one of the recognized obstacles for NLG.

To bypass the evaluation bottleneck, we exploited linguistics-enhanced evaluation methods such as METEOR and DAE. We find out that a more robust T2G module may help maximize the benefits of cycle training for NLG.

7 Acknowledgments

This work was partially supported by the European Union's Horizon Europe research and innovation programme within the ENEXA project (grant Agreement no. 101070305).

References

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*, pages 65–72, Michigan. ACL.

Bhuwan Dhingra, Manaal Faruqui, and et al. 2019. Handling divergent reference texts when evaluating table-to-text generation. In *Proceedings of the 57th Annual Meeting of the ACL*, pages 4884–4895, Florence, Italy. ACL.

Sebastian Gehrmann, Tosin Adewumi, and et al. 2021. The GEM benchmark: Natural language generation, its evaluation and metrics. In *Proceedings of the 1st Workshop on NLG, Evaluation, and Metrics*, pages 96–120, online. ACL.

Clément Godard, Oisin Mac Aodha, and Gabriel J Brostow. 2017. Unsupervised monocular depth estimation with left-right consistency. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 270–279.

Tanya Goyal and Greg Durrett. 2020. Evaluating factuality in generation with dependency-level entailment. In *Findings of the ACL: EMNLP 2020*, pages 3592–3603, Online. ACL.

Tanya Goyal and Greg Durrett. 2021. Annotating and modeling fine-grained factuality in summarization. In *Proceedings of the 2021 NAACL: HLTs*, pages 1449–1462, Online. ACL.

Qipeng Guo, Zhijing Jin, and et al. 2020. CycleGT: Unsupervised graph-to-text and text-to-graph generation via cycle training. In *Proceedings of the 3rd International Workshop on NLG from the Semantic Web*, pages 77–88, Dublin, Ireland (Virtual). ACL.

Yangfeng Ji and et al. 2020. The amazing world of neural language generation. In *Proceedings of the 2020 Conference on EMNLP: Tutorial Abstracts*, pages 37–42, Online. ACL.

Ziwei Ji, Nayeon Lee, Rita Frieske, Tiezheng Yu, and et al. 2022. Survey of hallucination in natural language generation. *ACM Comput. Surv.* Just Accepted.

Yuanmin Leng and et al. 2020. Controllable neural nlg: comparison of sota control strategies. In *Proceedings of the 3rd International Workshop on NLG from the Semantic Web*, pages 34–39, Virtual. ACL.

Chin-Yew Lin. 2004. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81, Bcn, Spain. ACL.

Pablo N Mendes and et al. 2011. Dbpedia spotlight: shedding light on the web of documents. In *Proceedings of the 7th international conference on semantic systems*, pages 1–8.

Kishore Papineni and et al. 2002. Bleu: A method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting on ACL*, ACL '02, page 311–318, USA. ACL.

Peng Qi and et al. 2020. Stanza: A python natural language processing toolkit for many human languages. In *Proceedings of the 58th Annual Meeting of the ACL: System Demonstrations*, pages 101–108, Online. ACL.

Colin Raffel, Noam Shazeer, and et al. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of ML Research*, 21(140):1–67.

Leonardo F. R. Ribeiro, Martin Schmitt, and et al. 2021. Investigating pretrained language models for graph-to-text generation. In *Proceedings of the 3rd Workshop on NLP for Conversational AI*, pages 211–227, Online. ACL.

Martin Schmitt, Sahand Sharifzadeh, Volker Tresp, and Hinrich Schütze. 2020. An unsupervised joint system for text generation from knowledge graphs and semantic parsing. In *Proceedings of the 2020 EMNLP*, pages 7117–7130, Online. ACL.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Improving neural machine translation models with monolingual data. In *Proceedings of the 54th Annual Meeting of the ACL (Volume 1: Long Papers)*, pages 86–96, Berlin, Germany. ACL.

Bayu Distiawan Trisedya and et al. 2019. Neural relation extraction for knowledge base enrichment. In *Proceedings of the 57th Annual Meeting of ACL*, pages 229–240, Florence, Italy. ACL.

Thomas Wolf and et al. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on EMNLP: System Demonstrations*, pages 38–45, Online. ACL.

Giulio Zhou and Gerasimos Lampouras. 2020. WebNLG challenge 2020: Language agnostic delexicalisation for multilingual RDF-to-text generation. In *Proceedings of the 3rd International Workshop on NLG from the Semantic Web*, pages 186–191, online. ACL.