Actes de CORIA-TALN 2023
Paris, France, 5 au 9 juin 2023

Evaluating the Generalization Property of Prefix-based Methods for Data-to-text Generation

Clarine Vongpaseut[1, 2] Alberto Lumbreras [1] Mike Gartrell[1] Patrick Gallinari[1, 3]
(1) Criteo AI Lab, Paris, France
(2) École des Ponts ParisTech, Champs-sur-Marne, France
(3) Sorbonne Universite, CNRS, ISIR, Paris, France
clarinevong@gmail.com, a.lumbreras@criteo.com, m.gartrell@criteo.com,
patrick.gallinari@sorbonne-universite.fr

RÉSUMÉ

Évaluation de la capacité de généralisation de méthodes *prefix-based* pour le *data-to-text*

Le *fine-tuning* est le paradigme courant pour adapter des modèles de language pré-entrainés à une tâche. Les méthodes de *fine-tuning* léger, comme le *prefix-tuning*, modifient uniquement un petit ensemble de paramètres, permettant de réduire les coûts d'entrainement. Ces méthodes atteignent des résultats comparables au *fine-tuning*. Toutefois, leurs performances se dégradent lorsqu'on s'éloigne des données d'entraînement. De plus, des travaux récents questionnent l'efficacité de ces méthodes selon la tâche d'application et la taille du modèle. Nous proposons dans ce papier d'évaluer la capacité de généralisation de méthodes *prefix-based* en fonction de la taille du modèle pré-entrainé, dans le cadre multi-domaine pour le *data-to-text* i.e. la conversion de données structurées en texte. Nous observons que leurs performances dépendent fortement de la taille du modèle.

ABSTRACT

Fine-tuning is the prevalent paradigm to adapt pre-trained language models to downstream tasks. Lightweight fine-tuning methods, such as prefix-tuning, only tune a small set of parameters which alleviates cost. Such methods were shown to achieve results similar to fine-tuning ; however, performance can decrease when the inputs get farther from the training domain. Moreover, latest works questioned the efficiency of recent lightweight fine-tuning techniques depending on the task and the size of the model. In this paper, we propose to evaluate the generalization property of prefix-based methods depending on the size of the pre-trained language model in the multi-domain setting on data-to-text generation. We found that their performance depends heavily on the size of the model.

MOTS-CLÉS : *Prefix-tuning*, Apprentissage multi-tâche, Capacité de généralisation, *Data-to-text*.

KEYWORDS: Prefix-tuning, Multi-task learning, Generalization property, Data-to-text.

1 Introduction

Fine-tuning is the prevalent approach to adapting pre-trained language models (PLMs) to various downstream tasks (Devlin *et al.*, 2019). However, fine-tuning is both computationally and memory expensive. Lightweight fine-tuning methods address this problem by freezing most of the PLMs parameters, e.g. fine-tuning the top layers, or by training only a smaller set of added parameters. A first method, adapter-tuning (Houlsby *et al.*, 2019), inserts task-specific layers between the layers of PLMs. Recently Li & Liang (2021) presented a second method, prefix-tuning, where a prompt

Actes de la 30e Conférence sur le Traitement Automatique des Langues Naturelles (TALN), volume 2 : travaux de recherche originaux – articles courts, pages 73–81.

is optimized as hidden-states, i.e. key-value for Transformer (Vaswani *et al.*, 2017). Their method outperformed adapter-tuning and also achieved comparable performance with fine-tuning on data-to-text benchmarks when using GPT-2 medium and large (Radford *et al.*, 2019). Lester *et al.* (2021) introduce a third lightweight fine-tuning method, prompt-tuning, in which continuous embeddings are optimized as soft prompts. Their experiments show that the performance gap between prompt-tuning and fine-tuning reduces with increase of model's size. In this paper, we focus on prefix-based methods.

Lightweight fine-tuning methods are also used for multi-task learning. HyperFormer (Mahabadi *et al.*, 2021) builds on adapter-tuning and uses hypernetworks to generate task and layer-specific adapter parameters, conditioned on task and layer embeddings. When published, HyperFormer achieved better performance on average on the GLUE benchmark (Wang *et al.*, 2018) compared to fine-tuning and adapter-tuning, when using T5-small and T5-base (Raffel *et al.*, 2020). He *et al.* (2022) propose HyperPrompt, a multi-task method based on prefix-tuning and HyperFormer's work, which outperforms HyperFormer. They showed that only tuning the added parameters of HyperFormer and HyperPrompt did not lead to similar performance when compared to tuning both added parameters and the PLM's parameters on SuperGLUE (Wang *et al.*, 2019), a more difficult NLU benchmark, when using T5-large. This observation questions the efficiency of lightweight fine-tuning since the performance of those methods seem to vary depending on the task difficulty. Clive *et al.* (2022) concatenate two prefixes : one for the task and one for the domain.

We consider here data-to-text generation tasks, consisting in generating fluent descriptions of data available in table or graph format and typically extracted from databases. We consider a challenging multi-domain setting where data is supposed to come from multiple sources, each with its own term and term-relations distributions. This could be considered as a specific multi-task problem where each domain corresponds to a task. The challenge here is to adapt PLMs in order to handle this multi-domain setting when current practice considers mono-domain settings. We then propose to evaluate the generalization property of prefix-based methods in the multi-domain setting, both in zero-shot and after few-shot fine-tuning on new target domains. With the question of efficiency depending on the model's size in mind, we compare results for T5-small and T5-base transformers.

2 Prefix-based methods

2.1 Prefix-tuning

In this section, we detail prefix-tuning (Li & Liang, 2021). We consider a PLM with frozen parameters ϕ. We use E, D and D_c to denote the three classes of attention present in each layer respectively, for the self-attention in the encoder and decoder and cross-attention in the decoder.

For each attention class, a set of prependable key-value pairs is learnt $P = \left\{(P_k^i, P_v^i)\right\}_{1 \leq i \leq N}$ with $P_{k,v}^i \in \mathbb{R}^{l \times h \times d_h}$, where N is the number of transformer blocks, l is the length of the prefix, h is the number of attention heads and d_h is the dimension of each head.

At the i-th transformer block, the additional key-value pairs are concatenated to the original key and value matrices, denoted K^i, V^i : $\quad K^{i\prime} = \left[P_k^i, K^i\right] \quad V^{i\prime} = \left[P_v^i, V^i\right]$. Multi-head attention is then computed using the new key-value and the original query Q^i.

The overall prefix, parameterized by θ, is denoted $P_\theta = \left\{P^E, P^D\ P^{D_c}\right\}$ and optimized through gradient descent : $\max_\theta \sum_{j=1}^{n} \log P(Y_j | X_j, P_\theta, \phi)$.

The prefix is not optimized directly. For each attention class, we learn $W \in \mathbb{R}^{l \times d}$ with d the model's

dimension and a two-layered feed-forward network with a bottleneck architecture, which takes as input W and outputs the key-value pairs for all transformer blocks $P = \left\{(P_k^i, P_v^i)\right\}_{1 \leq i \leq N}$. After training, the output of the MLPs can be saved and their weights can be dropped.

$$P_\theta = \left\{P^E,\, P^D\, P^{D_c}\right\} = \left\{MLP^E(W^E),\, MLP^D(W^D),\, MLP^{D_c}(W^{D_c})\right\}.$$

2.2 HyperPrompt

In this section, we present HyperPrompt (He *et al.*, 2022) for multi-domain learning. We consider a set of M domains $S_{train} = \{S^\tau\}_{1 \leq \tau \leq M}$ where $S^\tau = \{(X_j^\tau, Y_j^\tau)\}_{1 \leq j \leq n_\tau}$ is the τ-th domain and a PLM with frozen parameters ϕ. We introduce domain-conditioned parameters $\{\theta_\tau\}_{1 \leq \tau \leq M}$, which are optimized through gradient descent $\max_{\{\theta_\tau\}_{1 \leq \tau \leq M}} \sum_{\tau=1}^{M} \sum_{j=1}^{n_\tau} \log P(Y_j^\tau | X_j^\tau, \theta_\tau, \phi)$.

In HyperPrompt, domain-specific information is contained in hyperprompts, which are prependable key-value pairs only used in the self-attention of both the encoder and the decoder. Re-using previous notations, at the i-th transformer block, the original key-values K^i, V^i of the self-attention are augmented : $\quad K^{i\prime} = [P_{k,\tau}^i,\, K^i] \quad V^{i\prime} = \left[P_{v,\tau}^i,\, V^i\right]$

The different architectures to generate the hyperprompts are based on Mahabadi *et al.* (2021)'s work. For each domain, we learn a global prompt $W_\tau \in \mathbb{R}^{l \times d}$, a matrix containing domain-specific information, where l is the length of the prompt and d is the hidden size of the PLM's layers.

For each transformer block i, we have two local hypernetworks h_k^i and h_v^i that take a global prompt W_τ and output layer-specific and task-specific key-value.

$$P_{k,\tau}^i = h_k^i(W_\tau) = U_k^i(\sigma(D_k^i(W_\tau))) \tag{1}$$
$$P_{v,\tau}^i = h_v^i(W_\tau) = U_v^i(\sigma(D_v^i(W_\tau))) \tag{2}$$

The hypernetworks have a bottleneck architecture where $D_{k,v}^i \in \mathbb{R}^{d \times b}$ (resp. $U_{k,v}^i \in \mathbb{R}^{b \times h \times d_h}$) denotes the down-projection matrix (resp. the up-projection matrix), b is the bottleneck dimension and σ is a non-linear activation function.

HyperPrompt-Share In this method, at each transformer block i, all domains share the same two local hypernetworks h_k^i and h_v^i.

HyperPrompt-Separate In this method, at each transformer block i, each domain has its two own local hypernetworks h_k^i and h_v^i. Each domain-specific hyperprompt is trained independently, no knowledge is shared between domains.

HyperPrompt-Global In this method, we learn a task embedding $k_\tau \in \mathbb{R}^{t'}$ for each domain and a layer embedding $z_i \in \mathbb{R}^{t'}$ for each layer i. A projection network combines both layer and task embeddings into a layer-aware task embedding : $I_\tau^i = h(k_\tau, z_i) \in \mathbb{R}^t$. All tasks and all transformer blocks share two global hypernetworks H_k and H_v, which project I_τ^i into the weight matrices of the local hypernetworks (eqs. 1 and 2) :

$$(U_{k,\tau}^i, D_{k,\tau}^i) = H_k(I_\tau^i) = (W^{U_k}, W^{D_k}) I_\tau^i \tag{3}$$
$$(U_{v,\tau}^i, D_{v,\tau}^i) = H_v(I_\tau^i) = (W^{U_v}, W^{D_v}) I_\tau^i \tag{4}$$

2.3 HyperPrefix

Inspired by the two previous approaches, we propose to introduce a new multi-domain learning model, HyperPrefix. The domain-specific information is also contained in key-value pairs, which are prepended to original key, value matrices. The difference with HyperPrompt is how the prefix/hyperprompt is generated, here we use the same architecture as prefix-tuning.

For each attention class E, D and D_c, we learn a global prompt $W_\tau \in \mathbb{R}^{l \times d}$. All domains share the same hypernetwork H, a two-layered feed-forward network with a bottleneck architecture, which takes as input W_τ and generates key-value pairs for all transformer blocks.

$$P_\tau = \left\{(P^i_{k,\tau}, P^i_{v,\tau})\right\}_{1 \leq i \leq N} = H(W_\tau)$$

3 Experiments

3.1 Experimental setup

Pre-trained language models For our experiments, we use T5-small (60M parameters) and T5-base (220M parameters).

Models We compare fine-tuning, prefix-tuning, HyperPrefix and the different versions of HyperPrompt. Our implementation of the different models is based on Xie *et al.* (2022)'s implementation of prefix-tuning [1].

Datasets We evaluate the models on WebNLG datasets (Gardent *et al.*, 2017) (Shimorina & Gardent, 2018). WebNLG data consist of pairs of RDF triple sets (subject, predicate, object) and their associated reference text. The objective of data-to-text is then to generate the textual description associated to a set of triplets. WebNLG 2017 training samples are from 10 categories. In the test set, we have 5 additional unseen categories. We consider here each category as a domain (this is the closest instantiation of our multi-domain problem we have found in available data-to-text public benchmarks). We linearize the graph input, e.g. if the input is composed of two triples, as follows : $\text{subject}_1 : \text{predicate}_1 : \text{object}_1 \mid \text{subject}_2 : \text{predicate}_2 : \text{object}_2$

Metrics We report the following automatic metrics to evaluate the different models : BLEU (Papineni *et al.*, 2002), METEOR (Banerjee & Lavie, 2005) and TER (Snover *et al.*, 2006). We use (Post, 2018)'s implementation of BLEU and TER [2] and NLTK's implementation of METEOR [3].

Hyperparameters and training details All models are trained for 50 epochs, using Adafactor optimizer (Shazeer & Stern, 2018). The initial learning rate is set at 5e-5 and we use a linear learning rate scheduler. The batch size is set to 32. We apply early stopping based on the average development set metric. We use beam-search decoding with a beam size of 4. The prompt length l is set to 10. The bottleneck dimension b of all networks generating the prefix/prompt is $d/4$ where d the hidden size of the PLM's layers. For HyperPrompt-Global, the dimension of the task, layer and layer-aware task embeddings is 32, the hidden dimension of task-layer projection network h_t is 8.

1. https://github.com/hkunlp/unifiedskg
2. https://github.com/mjpost/sacrebleu
3. https://www.nltk.org/

3.2 Zero-shot learning

For the zero-shot experiments, we train the different models on WebNLG 2017. For prefix-tuning and fine-tuning, we treat the WebNLG 2017 dataset as a single domain. For the multi-domain learning models, we define each category as a domain. At testing time when the encountered category was not seen during training, the multi-domain models use the prefix/hyperprompt of the closest category. We use the Euclidean distance of GloVe embeddings (Pennington *et al.*, 2014) to measure the similarity between two categories as in (Clive *et al.*, 2022). We also test the importance of the prefix/hyperprompt in the decoder cross-attention by doing an ablation study. The scores are reported in Table 1.

	WebNLG								
	BLEU			METEOR			TER↓		
	S	U	A	S	U	A	S	U	A
SOTA (T5-large fine-tuned)	*65.82*	*56.01*	*61.44*	-	-	-	-	-	-
T5-small									
Fine-tuning	**62.93**	**44.60**	**54.83**	**63.52**	**53.55**	**58.75**	**52.86**	67.39	**59.45**
Prefix-tuning ✦	50.78	41.37	46.62	55.48	50.06	52.89	58.76	**65.22**	61.69
Prefix-tuning	49.94	40.32	45.69	54.56	49.19	51.99	58.79	65.31	61.75
HyperPrefix ✦	53.35	37.84	46.51	56.97	46.96	52.18	56.92	67.56	61.75
HyperPrefix	51.45	37.13	45.17	54.98	45.47	50.43	57.99	66.55	61.87
HyperPrompt-Global ✦	57.41	33.40	46.93	59.19	42.80	51.35	55.13	71.21	62.41
HyperPrompt-Global	54.53	32.29	44.83	57.20	43.08	50.45	56.03	69.76	62.25
HyperPrompt-Share ✦	54.01	34.02	45.28	56.14	43.47	50.07	56.68	70.01	62.73
HyperPrompt-Share	52.14	33.11	43.84	55.22	42.48	49.13	57.22	69.30	62.70
HyperPrompt-Sep ✦	55.79	30.15	44.60	57.29	40.40	49.20	56.63	74.39	64.68
HyperPrompt-Sep	54.70	29.36	43.71	56.55	39.84	48.55	56.58	73.56	63.82
T5-base									
Fine-tuning	**64.22**	**48.86**	**57.42**	**64.04**	55.04	**59.73**	**52.04**	63.56	**57.26**
Prefix-tuning ✦	-	-	-	-	-	-	-	-	-
Prefix-tuning	60.16	48.60	55.01	62.55	**55.36**	59.11	53.10	62.79	57.49
HyperPrefix ✦	-	-	-	-	-	-	-	-	-
HyperPrefix	62.01	47.70	55.67	62.34	54.52	58.59	52.89	**62.56**	57.28
HyperPrompt-Share ✦	61.15	41.09	52.30	61.55	51.19	56.59	53.41	66.74	59.45
HyperPrompt-Share	60.35	41.70	52.15	60.63	51.27	56.15	54.70	66.36	59.99
HyperPrompt-Sep ✦	58.55	37.72	49.43	58.84	47.32	53.33	55.73	67.69	61.15
HyperPrompt-Sep	58.23	37.03	48.90	59.27	46.80	53.30	55.92	68.81	61.76

TABLE 1 – Results on WebNLG test set, best results are marked in bold. T5-large fine-tuned results are from (Ribeiro *et al.*, 2021). ✦ indicates that we add prefix/prompt in all attention classes vs only in the self-attention. S, U and A refer to scores for the *seen*, *unseen* and *all* portions of the test set

Overall, the fine-tuned PLMs achieve the best results. The performance of prefix-based methods seems to heavily depend on the model's size. For T5-small, the scores of prefix-based methods are way below the results of T5-small fine-tuned. This difference between fine-tuning and prefix-based gets smaller when using T5-base, we go from a gap of between 8 and 11 pts of BLEU on the whole test set to a difference between 2 and 9 pts. These results point in the same direction as (Lester *et al.*, 2021), where the performance gap between prompt-tuning and fine-tuning narrows as the size of the model grows.

When comparing the different HyperPrompt methods, we can see that HyperPrompt-Global performs the best. Hyperprompt-Separate, on the other side, performs the worst, which is expected since no knowledge is shared between domains. Among the multi-domain learning methods, HyperPrefix works best on new domains. We note that all multi-domain models hallucinate and generate words linked to the mapped category. Qualitatively, prefix-based and fine-tuned models are able to copy subjects and objects that were not seen during training, but they struggle to correctly transcribe new relations when used in the zero-shot manner (examples in Figure 1).

Adding prefix/hyperprompt in the decoder cross-attention module led to a slight score improvement

Input : Ace Wilder : genre : Hip hop music | Hip hop music : stylistic_origin : Disco **Reference text** : Ace Wilder's musical genre is Hip hop music which has its origins in Disco.
HyperPrefix : Ace Wilder is a character in hip hop music which is stylistically influenced by disco. **Fine-tuned T5-base** : Ace Wilder is a member of the genre Hip Hop music, whose stylistic origin is Disco.

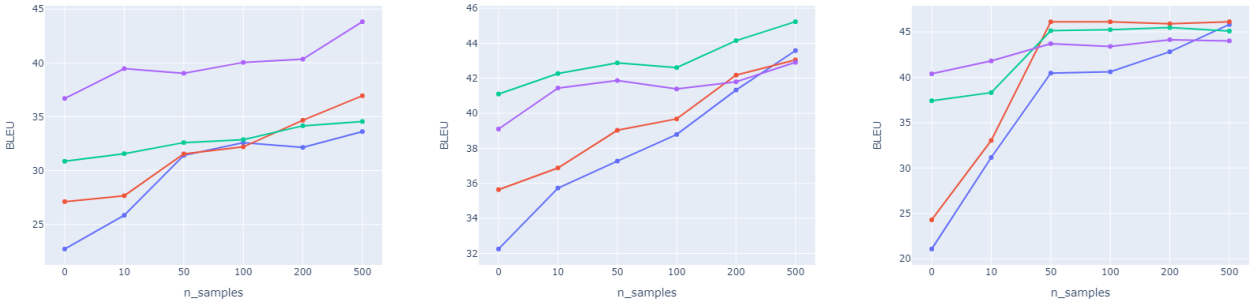
FIGURE 1 – Hallucinations produced by models with T5-base for a sample from the *Artist* category, which is not seen during training. Hallucination linked to the mapped category *ComicsCharacter* is highlighted in blue. Hallucinations linked to new predicates are emphasized in red.

with T5-small, whereas it is not significant for T5-base. In our following experiments, we only prepended prefix/hyperprompt in the self-attention module.

3.3 Few-shot domain adaptation

Since the tested models have good performance on seen domains, we evaluate if an already trained model on a set of domains can generalize to a new domain in the few-shot regime.

We consider a new category $C \in \{Artist, CelestialBody\}$ and subsample various number of examples (10, 50, 100, 200, 500) from WebNLG 2020 training set to constitute our training data. We use as validation set (resp. test set) the subset of all samples of category C from the WebNLG2020 validation dataset (resp. test set). We evaluate our models trained on WebNLG 2017 on the new domain after few-shot fine-tuning. For HyperPrompt-Share and HyperPrefix, we initialize the global prompt W_C with the global prompt of the closest category.



(a) *Artist*, T5-small (b) *Artist*, T5-base (c) *CelestialBody*, T5-base

FIGURE 2 – Few-shot domain transfer results : BLEU scores (avg. across 2 runs) of Hyperprompt-Share, HyperPrefix, prefix-tuning and fine-tuning models on two new domains *Artist* and *CelestialBody* after few-shot fine-tuning vs. # of training samples (0, 10, 50, 100, 200, 500)

We can observe in Figures 2(a) and 2(b) a difference in behaviour depending on the model's size. When using T5-small, the performance of prefix-based methods are way below regular fine-tuning. On the other hand, the BLEU scores of HyperPrompt-Share, HyperPrefix and prefix-tuning based on T5-base are able to reach fine-tuned T5-base when the number of samples for few-shot fine-tuning gets bigger.

4 Conclusion

Our study of prefix-based methods for Transformer shows that their performance depends heavily on the PLM's size. Scores of best prefix-based models were only comparable to fine-tuning for both trained and new categories when using T5-base. We also observe a difference of behaviour linked to the model's size for few-shot domain adaptation, where prefix-based models are able to reach results of fine-tuning as the number of training samples increases only with T5-base.

In the context of saving memory, additional experiments on few-shot domain adaption could be done : we could freeze the hypernetworks in HyperPrefix and HyperPrompt-Share and only train the new global prompts associated with the new domains. This type of few-shot domain adaption would be interesting since we would only need to save an additional matrix of dimension $(l \times d)$.

References

BANERJEE S. & 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*, p. 65–72, Ann Arbor, Michigan : Association for Computational Linguistics.

CLIVE J., CAO K. & REI M. (2022). Control prefixes for parameter-efficient text generation. In *Proceedings of the 2nd Workshop on Natural Language Generation, Evaluation, and Metrics (GEM)*, p. 363–382, Abu Dhabi, United Arab Emirates (Hybrid) : Association for Computational Linguistics.

DEVLIN J., CHANG M.-W., LEE K. & TOUTANOVA K. (2019). Bert : Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics : Human Language Technologies, Volume 1 (Long and Short Papers)*, p. 4171–4186.

GARDENT C., SHIMORINA A., NARAYAN S. & PEREZ-BELTRACHINI L. (2017). Creating training corpora for nlg micro-planners. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1 : Long Papers)*, p. 179–188 : Association for Computational Linguistics. DOI : 10.18653/v1/P17-1017.

HE Y., ZHENG S., TAY Y., GUPTA J., DU Y., ARIBANDI V., ZHAO Z., LI Y., CHEN Z., METZLER D., CHENG H.-T. & CHI E. H. (2022). HyperPrompt : Prompt-based task-conditioning of transformers. In K. CHAUDHURI, S. JEGELKA, L. SONG, C. SZEPESVARI, G. NIU & S. SABATO, Éds., *Proceedings of the 39th International Conference on Machine Learning*, volume 162 de *Proceedings of Machine Learning Research*, p. 8678–8690 : PMLR.

HOULSBY N., GIURGIU A., JASTRZEBSKI S., MORRONE B., DE LAROUSSILHE Q., GESMUNDO A., ATTARIYAN M. & GELLY S. (2019). Parameter-efficient transfer learning for nlp. In *International Conference on Machine Learning*, p. 2790–2799 : PMLR.

LESTER B., AL-RFOU R. & CONSTANT N. (2021). The power of scale for parameter-efficient prompt tuning. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, p. 3045–3059.

LI X. L. & LIANG P. (2021). Prefix-tuning : Optimizing continuous prompts for generation. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1 : Long Papers)*, p. 4582–4597.

MAHABADI R. K., RUDER S., DEHGHANI M. & HENDERSON J. (2021). Parameter-efficient multi-task fine-tuning for transformers via shared hypernetworks. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1 : Long Papers)*, p. 565–576.

PAPINENI K., ROUKOS S., WARD T. & ZHU W.-J. (2002). Bleu : a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, p. 311–318, Philadelphia, Pennsylvania, USA : Association for Computational Linguistics. DOI : 10.3115/1073083.1073135.

PENNINGTON J., SOCHER R. & MANNING C. D. (2014). Glove : Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, p. 1532–1543.

POST M. (2018). A call for clarity in reporting BLEU scores. In *Proceedings of the Third Conference on Machine Translation : Research Papers*, p. 186–191, Belgium, Brussels : Association for Computational Linguistics.

RADFORD A., WU J., CHILD R., LUAN D., AMODEI D. & SUTSKEVER I. (2019). Language models are unsupervised multitask learners.

RAFFEL C., SHAZEER N., ROBERTS A., LEE K., NARANG S., MATENA M., ZHOU Y., LI W. & LIU P. J. (2020). Exploring the limits of transfer learning with a unified text-to-text transformer. *The Journal of Machine Learning Research*, **21**(1), 5485–5551.

RIBEIRO L. F., SCHMITT M., SCHÜTZE H. & GUREVYCH I. (2021). Investigating pretrained language models for graph-to-text generation. In *Proceedings of the 3rd Workshop on Natural Language Processing for Conversational AI*, p. 211–227.

SHAZEER N. & STERN M. (2018). Adafactor : Adaptive learning rates with sublinear memory cost. In *International Conference on Machine Learning*, p. 4596–4604 : PMLR.

SHIMORINA A. & GARDENT C. (2018). Handling rare items in data-to-text generation. In *Proceedings of the 11th International Conference on Natural Language Generation*, p. 360–370 : Association for Computational Linguistics.

SNOVER M., DORR B., SCHWARTZ R., MICCIULLA L. & MAKHOUL J. (2006). A study of translation edit rate with targeted human annotation. In *Proceedings of the 7th Conference of the Association for Machine Translation in the Americas : Technical Papers*, p. 223–231, Cambridge, Massachusetts, USA : Association for Machine Translation in the Americas.

VASWANI A., SHAZEER N., PARMAR N., USZKOREIT J., JONES L., GOMEZ A. N., KAISER Ł. & POLOSUKHIN I. (2017). Attention is all you need. *Advances in neural information processing systems*, **30**.

WANG A., PRUKSACHATKUN Y., NANGIA N., SINGH A., MICHAEL J., HILL F., LEVY O. & BOWMAN S. (2019). Superglue : A stickier benchmark for general-purpose language understanding systems. *Advances in neural information processing systems*, **32**.

WANG A., SINGH A., MICHAEL J., HILL F., LEVY O. & BOWMAN S. (2018). Glue : A multi-task benchmark and analysis platform for natural language understanding. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP : Analyzing and Interpreting Neural Networks for NLP*, p. 353–355.

XIE T., WU C. H., SHI P., ZHONG R., SCHOLAK T., YASUNAGA M., WU C.-S., ZHONG M., YIN P., WANG S. I., ZHONG V., WANG B., LI C., BOYLE C., NI A., YAO Z., RADEV D., XIONG C., KONG L., ZHANG R., SMITH N. A., ZETTLEMOYER L. & YU T. (2022). UnifiedSKG :

Unifying and multi-tasking structured knowledge grounding with text-to-text language models. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, p. 602–631, Abu Dhabi, United Arab Emirates : Association for Computational Linguistics.