Reproducibility report: Unifying Vision-and-Language Tasks via Text Generation

Doğukan Arslan
Department of Computer Engineering
Istanbul Technical University
arslan.dogukan@itu.edu.tr

Muratcan Ünsal
Department of Computer Engineering
Istanbul Technical University
unsalmur20@itu.edu.tr

Reproducibility Summary

This report includes reproducibility evaluation of the paper called Unifying Vision-and-Language Tasks via Text Generation.

Scope of Reproducibility

Main claims of the authors that we try to reproduce are basically whether their unified model achieves good performance nearly as existing state-of-the-art task-specific models.

Methodology

Since training from stracth requires a lot of computational power, we only tested pre-trained models by authors. Also we make use of provided inference codes.

Results

For the experiments that we are successfully conducted, most of the time we manage to replicate the original results.

What was easy

Since author's provide an excellent code repository with great examples it is easy to use pre-trained models to test their claims.

What was difficult

Some experiments could not be verified due to their high demand of computational power.

Communication with original authors

The authors have not been contacted during this study.

1 Introduction

Unifying Vision-and-Language Tasks via Text Generation (1) is an article authored by Jaemin Cho, Jie Lei, Hao Tan, and Mohit Bansal from UNC-Chapel Hill. In this work, they focus on the problem of the task-specific architecture and objective requirement of vision-and-language models. To remove this necessity they proposed a one-for-all framework that can learn how to generalize for different tasks with the same architecture. They claim that their model can generate text labels nearly as good as recently developed state-of-the-art vision-and-language models, for a given textual and visual input.

In order to test author's claims, provided code, pretrained models and the data is used. Rest of the report is structured as follows: in Section 2 extent of the reproducibility is discussed, in Section 3 our methods to test main claims of the authors are detailed, in Section 4 reproducibility results are shared, and finally in Section 5 results and reproducibility process is discussed.

2 Scope of reproducibility

Corcerning to reproducibility of the article, not all claims are taken into consideration due to some restrictions such as huge training cost and GPU power. Hence, instead of training the model from scratch pretrained models which are shared by the authors are used.

Here are the claims that in the scope of this reproducibility report:

- Model can learn different tasks (to generate text labels) based on the visual and textual inputs.
- Model achieves performance nearly as good as recently developed state-of-the-art vision-and-language models.

With experiments in Section 4, we tested these claims.

3 Methodology

In our experiments we have used the official Github repository of the author's work and we have coded up 7 notebooks for the downstream task experiments with the same structure. These notebook links are provided in Section 3.3. Paper has a well written code base with a helpful read me file. It shows the code structure and examples of API usage. Also it includes Google Drive links of the datasets and pretrained models, and the usage of the bash scripts for fine tuning. Finally, writers are provided a nice inference Jupyter Notebook to readers to visualize visual question answering tasks with their framework.

Due to lack of resources in Google Colab and our local computers, we experimented by using pretrained models VL-T5 and VL-Bart on downstream tasks without fine tuning.

3.1 Model descriptions

You can see the hyper-parameters of the original experiments for pretraining and downstream tasks in the Table 3. During training they have used mixed precision training (2). Batch size is selected as 320 for VL-T5 and 600 for VL-Bart. AdamW (3) is used as an optimizer with 1e-4 learning rate and %5 linear warm-up is utilized. Code base is in PyTorch (4) and uses HuggingFace Transformers library (5). During testing, we have selected batch size as 100 for all the experiments due to the lack of resources in Google Colab.

3.2 Datasets

3.2.1 Pretraining

Pre-training datasets and their sizes can be seen on Table 1. COCO[1] and Visual Genome[2] datasets are used as image sources for all the tasks. For visual question answering tasks, VQA[3], GQA[4] and Visual7W[5] datasets are utilized. In total there are 9.18 image-text pairs on 180K distinct images.

[1] https://cocodataset.org/home

[2] http://visualgenome.org

[3] https://visualqa.org

[4] https://cs.stanford.edu/people/dorarad/gqa/

[5] http://ai.stanford.edu/ yukez/visual7w/

Table 1: Pretrained tasks used in vision-and-language pretraining.

Task	Image Source	Text Source	# Examples
Multimodal language modelling	COCO, VG	COCO caption, VG caption	4.9M (# of captions)
Visual question answering	COCO, VG	VQA, GQA, Visual7W	2.5M (# of captions)
Image-text matching	COCO	COCO caption	533K (# of captions)
Visual grounding	COCO, VG	object&attribute tags	163K (# of captions)
Grounded captioning	COCO, VG	object&attribute tags	163K (# of captions)

Table 2: Statistics about fine tuning datasets.

	Train	**Validation**	**Test**
VQA	113287	5000	5000
GQA	943000	132062	12578
NLVR	86373	6982	6967
VCR	212923	26534	25263
RefCOCOg	42226	2573	5023
COCO	113287	5000	5000
Multi30K	29000	1014	1000, 1000, 1000

3.2.2 Fine-tuning

There are 7 fine tuning tasks such as VQA, GQA, NLVR[6], VCR[7], RefCOCOg[8][9], COCO, and Multi30K[10]. Detailed statistics about fine tuning datasets can be seen Table 2.

Table 3: Hyperparameters for pretraining and downstream tasks.

Model	Task	Learning rate	Batch size	Epochs
VL-T5	Pretraining	1e-4	320	30
	VCR Pretraining	5e-5	80	20
	VQA	5e-5	320	20
	GQA	1e-5	240	20
	NLVR	5e-5	120	20
	RefCOCOg	5e-5	360	20
	VCR	5e-5	16	20
	COCO Caption	3e-5	320	20
	Multi30K En-De	5e-5	120	20
VL-BART	Pretraining	1e-4	600	30
	VCR Pretraining	5e-5	120	20
	VQA	5e-5	600	20
	GQA	1e-5	800	20
	NLVR	5e-5	400	20
	RefCOCOg	5e-5	1200	20
	VCR	5e-5	48	20
	COCO Caption	3e-5	520	20
	Multi30K En-De	5e-5	320	20

[6] https://lil.nlp.cornell.edu/nlvr/

[7] https://visualcommonsense.com/

[8] https://github.com/lichengunc/MAttNetpre-computed-detectionsmasks

[9] https://github.com/lichengunc/refer

[10] https://github.com/multi30k/dataset

Table 4: VL-T5 and VL-Bart test run time in minutes.

	Tasks						
	VQA	***GQA***	***NLVR***	***RefCOCOg***	***VCR***	***COCO***	***Multi30K***
VL-T5	13	4.21	3.53	2.16	-	-	-
VL-Bart	16	5.18	1.43	1.27	-	-	-

Table 5: Final results.

	VL-T5	**VL-Bart**
VQA	67.92	54.78
GQA	59.1	49.4
NLVR	50.1	48.6
RefCOCOg	0	2.36
VCR	-	-
COCO	-	-
Multi30K	-	-

3.3 Experimental setup and code

Codes provided by authors are pretty self explanatory and the Github repository has excellent explanations on how to perform experiments and do inferences on some of the tasks. Due to the lack of resources we were only able to run experiments by using pre-trained models on the downstream tasks. For this purpose we have created below Google Colab notebooks for the specified downstream task. In these notebooks we prepare the environment, importing the test dataset and running the experiments for both of the pre-trained models VL-T5 and VL-Bart. You can find these notebooks in the links below.

- COCO Captioning
- VCR
- VQA
- GQA
- NLVR
- RefCOCOg

3.4 Computational requirements

Experiments in the paper run on quite a strong setup. They had access to 4 RTX 2080 GPUs, each GPU have 12 GB RAM. Authors stated that pre-training took 4 full days on this setup. In our experiments we had to use Google Colab with 12 GB RAM and Nvidia K80 GPU with 12 GB RAM. Due to lack of resources we were only able to run experiments by using pre-trained models. For each task, testing run time can be seen on Table 4.

4 Results

As we have stated before, due to the lack of resources we were not able to replicate the experiments. Instead, we experimented on downstream datasets by using pretrained VL-T5 and VL-Bart. Since it is a unified approach, we have assumed that we should get acceptable results on the tasks such as visual question answering, or COCO captioning. Our results can be seen on Table 5.

4.1 Results reproducing original paper

Based on the author's claims that are mentioned earlier, experiments that have been conducted by us are discussed in this section.

4.1.1 VQA

In the VQA experiment, we got the same results as the authors by using pre-trained VL-T5. However, there is significant drop on accuracy when VL-Bart is used.

4.1.2 GQA

In the GQA experiment, both pre-trained models perform lower than the fine-tuned models. VL-T5 outputs %1 less accuracy while VL-Bart outputs %10 less.

4.1.3 NLVR

In the NLVR experiment, both pre-trained models perform a lot less than the fine-tuned models, by %20 less.

4.1.4 RefCOCOg

In the RefCOCOg experiment, both pre-trained models' accuracy results are almost 0. This is interesting because RefCOCOg is a visual grounding task which is one of the pretraining tasks.

4.1.5 COCO captioning and VCR

We were not able to run the experiments for COCO captioning and VCR tasks. We encountered an error which has a open ticket in Google Colab's repository since 2019[11]. Even though we have tried a couple of workarounds, it did not work out.

4.2 Results beyond original paper

Apart from the claims of the authors, significant bias has been observed during the test phase of the pre-training models. Since it is a serious issue more experiments should be conducted to investigate model's bias against some specific parts of the society.

5 Discussion

Even though we could not conduct all experiments from the article due to lack of computational power, we think that we observed valuable results when we test the pre-trained model on different tasks.

5.1 What was easy

Base code which is provided by the authors is clearly written and easy to run. Also it is easy to plug a new task to the provided code. In addition, explanation of the experiments is easy to follow in the article and the provided read me file.

5.2 What was difficult

The most difficult part of the reproduction study is the amount of source needed. Since authors conduct this study with high amount of computational power, replicating this and training model from scratch was not possible.

5.3 Communication with original authors

The authors have not been contacted during this study because code document is well-written.

[11] https://github.com/googlecolab/colabtools/issues/510

References

[1] J. Cho, J. Lei, H. Tan, and M. Bansal, “Unifying vision-and-language tasks via text generation,” in *ICML*, 2021.

[2] P. Micikevicius, S. Narang, J. Alben, G. F. Diamos, E. Elsen, D. García, B. Ginsburg, M. Houston, O. Kuchaiev, G. Venkatesh, and H. Wu, “Mixed precision training,” *ArXiv*, vol. abs/1710.03740, 2018.

[3] I. Loshchilov and F. Hutter, “Decoupled weight decay regularization,” in *ICLR*, 2019.

[4] A. Paszke, S. Gross, S. Chintala, G. Chanan, E. Yang, Z. DeVito, Z. Lin, A. Desmaison, L. Antiga, and A. Lerer, “Automatic differentiation in pytorch,” 2017.

[5] T. Wolf, L. Debut, V. Sanh, J. Chaumond, C. Delangue, A. Moi, P. Cistac, T. Rault, R. Louf, M. Funtowicz, and J. Brew, “Huggingface’s transformers: State-of-the-art natural language processing,” *ArXiv*, vol. abs/1910.03771, 2019.