On-device Vision-Language Navigation

Dhruv Naik
Carnegie Mellon University
Pittsburgh, PA 15213
drn@andrew.cmu.edu

Saloni Mittal
Carnegie Mellon University
Pittsburgh, PA 15213
salonim@andrew.cmu.edu

Thomas Xu
Carnegie Mellon University
Pittsburgh, PA 15213
twx@andrew.cmu.edu

1 Motivation

To enable robots to smoothly navigate through realistic 3D visual environments using natural language has been a long-standing challenge. In vision-and-language navigation (VLN) tasks, an embodied agent should first interpret the instructions and then determine if the visual inputs along a path match the descriptions provided in the instructions. Given the extremely diverse nature of visual and language inputs, the generalization of VLN agents to unseen environments remains a challenge. There have been recent works in developing large visiolinguistic transformer-based models that are pretrained on large image-text pairs from the web and also use fine-tuning to improve performance (see Related Work below).

One such pretrained model released in 2021 is AirBert (1) which is trained on millions of VLN path-instruction (PI) pairs. They use BnB (a large scale VLN dataset created from AirBnb data) for pretraining and show that the AirBert model outperforms the state-of-the-art for for Room-to-Room (R2R) navigation and Remote Referring Expression (REVERIE) (2) benchmarks. For our project we propose to use this huge transformer-based model for a VLN task. Our main focus will be on applying different compression/distillation techniques to AirBert, in order to run it on a 2GB Jetson Nano.

2 Related Work

Vision-language navigation has had numerous advances in recent years. As mentioned above, one of those was the AirBert model which performed well on VLN tasks in simulated indoor environments. That work also introduced the BnB dataset, as mentioned previously, which can be utilized in the future for other VLN-related research. There has also been recent work in this area from industry; for example, Facebook Research recently developed VLN-BERT (3) : a visiolinguistic transformer-based model for VLN tasks. In addition to outperforming other contemporary models on certain tasks, they also showed that pretraining on image-text pairs scraped from the web followed by fine-tuning on embodied path-instruction data led to significantly improved performance. Various simulators have been previously used for VLN research; some notable examples include the Matterport Simulator and Habitat simulation platform.

There has been little prior work on **compressing large-scale visual-linguistic transformers** but many model compression techniques that are applied to large transformer-based models are mainly task and domain agnostic. In *Fang et al.* (4), the authors have used knowledge distillation techniques at various stages of the model as was first proposed in *Jiao et al.*(5) to facilitate training of smaller vision-language (VL) models. They train the student model to match the attention, hidden layer and classification distributions of the teacher model with a MLM learning objective. The proposed DistillVLM model significantly improves performance of small VL models for VL tasks such as image captioning and visual question answering over their non-distilled counterparts. A major challenge highlighted by the authors is the inconsistent regional visual tokens extracted from different detectors of teacher and student models, resulting in the misalignment of hidden representations and attention distributions. To address this problem, they perform visual token alignment by retraining

the Teacher by using the same region proposals from Student's detector while the features are from Teacher's own object detector.

More recently, a large survey paper (6) came out, discussing compression techniques for large-scale transformer models, doing a case study on BERT. It is a detailed analysis and comparison of various BERT compression methods and offer great insights into practical use of these methods and potential future directions.

3 Hypothesis

The current size of the pre-trained AirBert transformer model is 2.3 GB. Fine-tuning this model on a downstream task might result in additional layers and parameters getting added to the model, therefore increasing the model size further. To run the model on the device, we will also have to leave room for loading/reading input from the Matterport Simulator and language instructions. Given the Jetson Nano size is only 2GB, it appears a reduction of approximately 60% in the model size is an ideal target, to be able to run it smoothly on the device. We will start by using the knowledge distillation technique described in the DistillBert paper as our first experiment and then adapting more advanced methods.

4 Datasets and Evaluation Metrics

We will be evaluating our model on Room-to-Room (R2R) dataset based on the Matterport3D simulator. R2R contains human-annotated path instruction pairs in previously unseen buildings that are divided into training, seen and unseen validation, and unseen testing sets. We will use the standard VLN evaluation metrics, as described below:

- **Success rate (SR)** measures the percentage of selected paths that stop within 3m of the goal. In path selection this is our primary metric of interest.
- **Navigation error (NE)** measures the average distance of the shortest path from the last position in the selected path to the goal position.
- **Path length (PL)** measures the average length of the selected path.

5 Baseline Models

Table 1 contains the performance of state-of-the-art models on the R2R dataset (7) test set. The Airbert model has the highest success rate (SR) at 77%, 4% more than VLN-BERT, and 1st of the R2R leaderboard. The human success rate on this task is 86%.

Model	PL	NE	SR
Airbert (1)	687	2.69	77
VLN-BERT (3)	687	3.09	73
AuxRN (8)	41	3.24	71
EnvDrop (9)	687	3.26	69

Table 1: Navigation performance on R2R test set on the leaderboard

6 Hardware and Compute Requirements

The minimum required hardware for this project is simply the Jetson Nano and a computer capable of running the Matterport Simulator. Ideally, we would also have access to a powerful compute platform (via either GCP/AWS credits or a physical lab computer with a GPU) to be used for model training, running 3D simulations.

If we have time to complete some of the project stretch goals (see "Potential Extensions" section below), then additional hardware would be needed to convert the Jetson Nano into a VLN agent. In that case, we would require: 1 small camera, 2 wheels driven by DC motors, 1 caster wheel, 1 motor driver board, 1 9V battery, and a plastic chassis.

7 I/O

The input and output modalities required by this project (excluding potential stretch goals) are very minimal. We simply require the Jetson Nano to be able to communicate to a laptop, which can be achieved via SSH or a similar communication protocol. The model inputs, which will be images and other data from the Matterport Simulator along with corresponding language commands, will be sent from the laptop to the Jetson. The language commands will be typed by us, and will be given in the form of text strings. The model outputs (chosen navigation actions at each timestep) from the Jetson will similarly be communicated back to the laptop. If we complete the core project and begin to work on stretch goals, then additional I/O modalities will be required (see Section 9).

8 Potential Challenges

This project has a high dependence on being able to run and use the Matterport Simulator software smoothly. If we run into many roadblocks in this respect that hinder the progress of our project, then we will change the direction of the project from being a navigation task to any other task that uses both vision and language modalities. We would still be able to use our main idea of compressing a visiolinguistic based transformer model for that task.

9 Potential Extensions

One potential extension of our project will be to first perform zero/few shot evaluation of the pre-trained VLN model on out-of-domain data like different visual environments and then work on improving model generalization.

Another stretch goal would be building an embodied agent that can actually navigate in a small physical environment using our model. For that we will have to replace the simulator and come up with a proxy that can perform the same tasks as the Matterport simulator. for our environment. This will also require additional I/O modalities: a camera to take pictures of the environment, a physical body with wheels to carry out chosen navigation actions, and a motor driver board for the Jetson to actuate the wheels.

10 Potential Ethical Implications

We currently do not expect any significant ethical implications for this project. In general, agents equipped with advanced VLN capabilities could potentially be used in unethical applications, but this project only focuses on the academic exercise of running VLN models on edge devices in the first place.

11 Timeline and Milestones

- Lab 2-3 (Sep 30 - October 7)
  - Setup remote compute, local setup Simulator. Run Airbert model successfully on R2R dataset.
- Lab 4-5 (Oct 14-28)
  - Run experiments with Distillation, Quantization.
- Lab 6 (Nov 4)
  - Reduce model size and latency enough, to eventually run on Nano. Compare performance with baselines and related work.
- Lab 7-8 (Nov 14-18)
  - Run VLN model on Jetson Nano. Experiment with architecture specific tricks to optimize memory consumption, latency, performance.
- Lab 9 (Dec 2)

  - Estimating energy consumption, carbon calculation. Preparing final report, presentation.
- Final Report (Dec 10)
  - VLN model runs on Jetson Nano, with performance comparable to baselines
  - Mobile robot w/Jetson Nano can run the VLN model to navigate a simple lab environment (stretch goal)

References

[1] P.-L. Guhur, M. Tapaswi, S. Chen, I. Laptev, and C. Schmid, "Airbert: In-domain Pretraining for Vision-and-Language Navigation," 2021.

[2] Y. Qi, Q. Wu, P. Anderson, X. Wang, W. Y. Wang, C. Shen, and A. van den Hengel, "Reverie: Remote embodied visual referring expression in real indoor environments," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[3] A. Majumdar, A. Shrivastava, S. Lee, P. Anderson, D. Parikh, and D. Batra, "Improving vision-and-language navigation with image-text pairs from the web," 2020.

[4] Z. Fang, J. Wang, X. Hu, L. Wang, Y. Yang, and Z. Liu, "Compressing visual-linguistic model via knowledge distillation," *ArXiv*, vol. abs/2104.02096, 2021.

[5] X. Jiao, Y. Yin, L. Shang, X. Jiang, X. Chen, L. Li, F. Wang, and Q. Liu, "Tiny{bert}: Distilling {bert} for natural language understanding," 2020. [Online]. Available: https://openreview.net/forum?id=rJx0Q6EFPB

[6] P. Ganesh, Y. Chen, X. Lou, M. A. Khan, Y. Yang, H. Sajjad, P. Nakov, D. Chen, and M. Winslett, "Compressing large-scale transformer-based models: A case study on bert," 2021.

[7] P. Anderson, Q. Wu, D. Teney, J. Bruce, M. Johnson, N. Sünderhauf, I. Reid, S. Gould, and A. van den Hengel, "Vision-and-language navigation: Interpreting visually-grounded navigation instructions in real environments," 2018.

[8] F. Zhu, Y. Zhu, X. Chang, and X. Liang, "Vision-language navigation with self-supervised auxiliary reasoning tasks," 2020.

[9] H. Tan, L. Yu, and M. Bansal, "Learning to navigate unseen environments: Back translation with environmental dropout," 2019.