VALET: Vision-And-LanguagE Testing with Reusable Components

Eric Slyman*
Oregon State University
Corvallis, OR, USA
slymane@oregonstate.edu

Kushal Kafle
Adobe
San Jose, CA, USA
kushal.kafle@adobe.com

Scott Cohen
Adobe
San Jose, CA, USA
scott.cohen@adobe.com

Abstract

Vision-and-Language (ViL) modeling advancements have resulted in significant improvements to aggregate metric performance on a variety of tasks. However, this evaluation may not accurately reflect a model's capability to behave as intended by its creator, or according to the expectations of an end-user. Behavioral testing and sensemaking methods have been identified as effective for surfacing these errors in ViL models, but are limited in practice by their ability to scale to many examples and involved engineering requirements. In order to be practical for ViL tasks and suitable for organizational-level testing, these methods must scale to large sample sizes without requiring costly or repetitive engineering efforts for each individual test case. To address these challenges, we propose VALET, a system designed to rapidly develop scalable behavioral tests for ViL models that offers a high-level interface for non-technical users to perform testing that is supported by a modular system of interoperable components enabling expert users to extend and share testing environments more easily. We present a case study using VALET to evaluate a language-guided model's capability to count in zero-shot image classification.

1 Introduction

Large-scale pretrained Vision-and-Language (ViL) models [Chen et al., 2022, Radford et al., 2021, Jia et al., 2021] have shown significant progress in improving performance on tasks such as visual question answering [Goyal et al., 2017, Johnson et al., 2017], image captioning [Lin et al., 2014, Agrawal et al., 2019], and text-based image retrieval [Lin et al., 2014, Plummer et al., 2015]. However, the use of aggregate metrics to evaluate model performance on static test sets may not always accurately reflect a model's ability to behave as expected with respect to some critical capability. For example, it may be desirable for a model to be capable of understanding *spatial relationships* (*e.g.*, Q: "*What is above the counter*" A:"*Two frying pans*"). Behavioral testing [Ribeiro et al., 2020, Rahwan et al., 2019] and sensemaking frameworks [Cabrera et al., 2023b] offer a pathway to profile consistent predictive patterns around these capabilities – the model's behavior – and expose systemic errors obfuscated by aggregate metrics where model predictions are consistently misaligned with human semantic understanding, but do not trivially extend to the ViL setting (*e.g.* using text templates to generate targeted test cases) or require extensive human input to manually catalogue (mis)predictions.

Several prior works have studied behavior verification for AI models. CheckList [Ribeiro et al., 2020] automates scalable behavior tests for NLP models by allowing users to write text templates which target specific model capabilities, but a ViL extension is nontrivial due to the time and accuracy constraints of generating fine-grained realistic images matching the corresponding text template for a task. Human-driven sensemaking frameworks like AIFinnity [Cabrera et al., 2023b] can provide

*Work conducted while interning at Adobe

Preprint. Under review.

insights into model capabilities, but are limited in scale by the number of testable samples due to their focus on exploratory affordances with individual instances of data over support for scalable testing scenarios. This work introduces VALET, a visual interface enabling rapid construction and execution of scalable behavior tests for ViL models. Unlike other methods, VALET offers testing automation alongside a composable system of reusable testing components that can scale to thousands of samples without the need for redundant work between users. Concurrent to our work is Zeno [Cabrera et al., 2023a], which provides a general platform for evaluating the behavior of AI models but does not allow for the transformation and permutation of existing data to generate novel test cases, and does not offer tooling to define scalable test templates based on intermodal relationships and metadata.

2 VALET Methodology

VALET comprises a high-level visual user interface for designing tests and an extensible system of discrete interoperable testing components which may be conveniently composed to form arbitrary tests directly from the interface. By enabling the flexible integration of these components, VALET facilitates the creation of complex ViL behavior tests while ensuring scalability and ease of use.

User Interface. The VALET visual interface allows users to compose components into tests, enabling them to rapidly verify desired model behaviors. First, (Fig. 1 A) users define conceptually aligned categories of tests, specifying a name and test type[2] for each. They may also load test suites previously shared by other users. Next, (Fig. 1 B) they configure individual components to be used in the test, such as models to evaluate, metrics to run, and permutations to perform. Users can automatically generate numerous testing samples by constructing text templates that may incorporate image metadata and user-defined functions. Finally, (Fig. 1 C) users can execute subsets of their tests and get diagnostic feedback of their model(s) performance. In the event of surprising behavior, users can download a JSON document containing model predictions for further investigation.

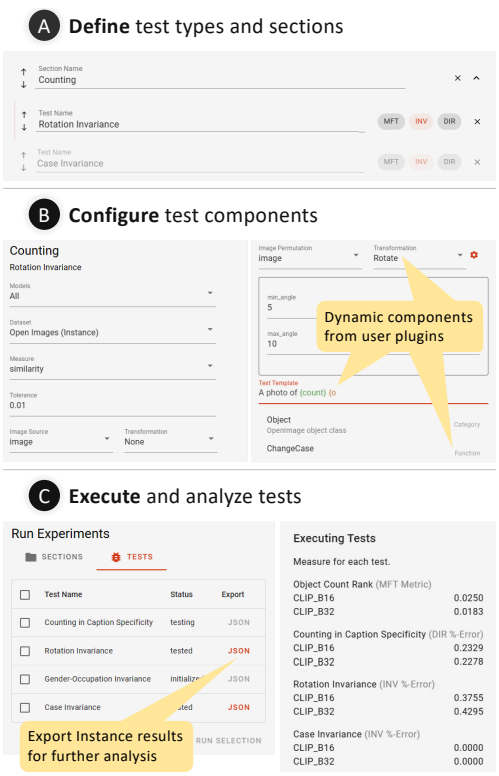

Figure 1: An example user workflow with VALET. A user begins by **(A) Defin**ing a conceptually aligned set of tests and sections then **(B) Configur**es each test within those sections before **(C) Executi**ng those tests and analyzing the results.

Reusable Components. VALET is based on a system of reusable Python components that can be easily shared among users, decomposing behavioral tests into four core elements: *datasets*, *permutations*, *measures*, and *models*, which offers technical users the flexibility to integrate new components without sacrificing interoperability. This modular architecture enables code reuse between tests, avoiding duplication of effort among users and creating a comprehensive ecosystem of testing components for use by non-technical users over time. As shown in Fig. 2, to construct a new component, a user must only extend the relevant parent class to perform a permutation and define metadata for configuration options to be exposed in the visual interface for use by non-technical users.

Unique to VALET is the ability to connect existing ViL datasets and image metadata into text templates when designing automatically scalable tests. A user may define functions to be used within templates that pull relevant information from an associated image (*e.g.*, the original caption, object counts, question-answer pairs) so that they may be leveraged for generating text (*e.g.*, new captions, permuted object counts, novel questions) which captures a true visiolinguistic relationship with an image.

[2]For simplicity, we use the same test types as Ribeiro et al. [2020]

3 Case Study: Counting

In this section, we detail an example scenario where a researcher uses VALET to diagnose a behavioral error in their model that was otherwise obfuscated by typical aggregate classification metrics.

A research scientist has developed a new ViL model to improve zero-shot classification accuracy in busy images. However, during testing, the new model shows no significant difference in accuracy to the baseline version on average. The scientist identifies *counting* as a critical capability of the model for their intended use case, particularly for identifying the number of subjects in an image (*e.g.*, "*A photo of four people*"). Improving this capability is known to increase customer satisfaction when interacting with the model. To evaluate this behavior, the scientist uses VALET to define a new section of *counting* tests (Fig. 1 Ⓐ) which they configure to measure a small number of counting-based samples to establish minimum functionality of the capability. Additionally, they automatically generate a large set of examples by writing a template (Fig. 1 Ⓑ) which permutes the text of counting-based prompts and masks counted objects in the image with known expected results. For example, masking all but one person from an image, and permuting the caption to "*A photo of one person*". Executing the tests (Fig. 1 Ⓒ) reveals that the original model makes a disparate number of errors concentrated in subject counting, whereas the new version does not. The researcher further analyzes specific examples of the test outputs to confirm this behavioral pattern, concluding that the new model is more desirable for their use case despite its similar average performance.

```
class MyPerm(valet.TextPermutation):
    def __init__(self):
        // initalize permutation

    def __call__(self, x):
        // permute packed data

    @classmethod
    def metadata(cls):
        // expose interface options
```

Figure 2: Example definition of a new TextPermutation component to include in the VALET interface. Data is shared between components such that text permutations may consider image metadata and vice versa. Allowing technical users to leverage multimodal relationships in their testing.

In this scenario, VALET enabled the researcher to uncover differences in the behavior of two models related to a critical capability that was not apparent from aggregate metrics, and may easily reuse their testing components in future evaluation. By constructing scalable tests for this behavior, the researcher verified critical capabilities and selected the model that better suits their customers' needs.

4 Conclusion

We present ongoing work on VALET, a tool for rapidly developing scalable behavior tests for Vision-and-Language models. VALET allows users to develop tests based on permutations of existing ViL datasets and by creating their own targeted samples based on templates which respect visiolinguistic relationships. We hope for VALET to allow stakeholders of varying technical ability to effectively evaluate the behavior of their models and identify persistent errors in critical capabilities.

References

Harsh Agrawal, Karan Desai, Yufei Wang, Xinlei Chen, Rishabh Jain, Mark Johnson, Dhruv Batra, Devi Parikh, Stefan Lee, and Peter Anderson. nocaps: novel object captioning at scale. *International Conference on Computer Vision*, pages 8947–8956, 2019. 1

Ángel Alexander Cabrera, Erica Fu, Donald Bertucci, Kenneth Holstein, Ameet Talwalkar, Jason I. Hong, and Adam Perer. Zeno: An interactive framework for behavioral evaluation of machine learning. In *CHI '23*. Association for Computing Machinery, 2023a. ISBN 9781450394215. doi: 10.1145/3544548.3581268. 2

Ángel Alexander Cabrera, Marco Tulio Ribeiro, Bongshin Lee, Robert Deline, Adam Perer, and Steven M Drucker. What did my ai learn? how data scientists make sense of model behavior. *ACM Transactions on Computer-Human Interaction*, 30(1):1–27, 2023b. 1

Xi Chen, Xiao Wang, Soravit Changpinyo, AJ Piergiovanni, Piotr Padlewski, Daniel Salz, Sebastian Alexander Goodman, Adam Grycner, Basil Mustafa, Lucas Beyer, Alexander Kolesnikov, Joan Puigcerver, Nan Ding, Keran Rong, Hassan Akbari, Gaurav Mishra, Linting Xue, Ashish Thapliyal, James Bradbury, Weicheng Kuo, Mojtaba Seyedhosseini, Chao Jia, Burcu Karagol Ayan, Carlos Riquelme, Andreas Steiner, Anelia Angelova, Xiaohua Zhai, Neil Houlsby, and Radu Soricut. Pali: A jointly-scaled multilingual language-image model. In *arXiv*, 2022. 1

Yash Goyal, Tejas Khot, Douglas Summers-Stay, Dhruv Batra, and Devi Parikh. Making the v in vqa matter: Elevating the role of image understanding in visual question answering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 6904–6913, 2017. 1

Chao Jia, Yinfei Yang, Ye Xia, Yi-Ting Chen, Zarana Parekh, Hieu Pham, Quoc V. Le, Yun-Hsuan Sung, Zhen Li, and Tom Duerig. Scaling up visual and vision-language representation learning with noisy text supervision. In *International Conference on Machine Learning (ICML)*, 2021. 1

Justin Johnson, Bharath Hariharan, Laurens Van Der Maaten, Li Fei-Fei, C Lawrence Zitnick, and Ross Girshick. Clevr: A diagnostic dataset for compositional language and elementary visual reasoning. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2901–2910, 2017. 1

Tsung-Yi Lin, Michael Maire, Serge J. Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. Microsoft coco: Common objects in context. In *European Conference on Computer Vision*, 2014. 1

Bryan A. Plummer, Liwei Wang, Christopher M. Cervantes, Juan C. Caicedo, J. Hockenmaier, and Svetlana Lazebnik. Flickr30k entities: Collecting region-to-phrase correspondences for richer image-to-sentence models. *International Journal of Computer Vision*, 123:74–93, 2015. 1

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. In Marina Meila and Tong Zhang, editors, *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 8748–8763. PMLR, 2021. 1

Iyad Rahwan, Manuel Cebrian, Nick Obradovich, Josh C. Bongard, Jean-François Bonnefon, Cynthia Lynn Breazeal, Jacob W. Crandall, Nicholas A. Christakis, Iain D. Couzin, Matthew O. Jackson, Nicholas R. Jennings, Ece Kamar, Isabel M. Kloumann, H. Larochelle, David Lazer, Richard Mcelreath, Alan Mislove, David C. Parkes, Alex 'Sandy' Pentland, Margaret E. Roberts, Azim F. Shariff, Joshua B. Tenenbaum, and Michael P. Wellman. Machine behaviour. *Nature*, 568: 477–486, 2019. 1

Marco Tulio Ribeiro, Tongshuang Wu, Carlos Guestrin, and Sameer Singh. Beyond accuracy: Behavioral testing of NLP models with CheckList. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4902–4912. Association for Computational Linguistics, 2020. doi: 10.18653/v1/2020.acl-main.442. 1, 2