Programming Language Representation with Semantic-level Structure

Anonymous Author(s)

ABSTRACT

NLP (NLP) technique becomes one of the core techniques for developing text analytics applications. For developing the NLP applications, the applications are required to achieve high reliability before it goes to market. The trustworthiness of the prevalent NLP applications is obtained by measuring the accuracy of the applications on held-out dataset. However, evaluating NLP on testset does with held-out accuracy is limited to show its quality because the held-out datasets are often not comprehensive. While the behavioral testing over multiple general linguistic capabilities are employed, it relies on manually created test cases, and is still limited to measure its comprehensive performance for each linguistic capability. In this work, we introduce Fuzz-CHECKLIST, an NLP model testing methodology. Given a linguistic capability, The Fuzz-CHECKLIST finds relevant testcases to test the linguistic capability from existing datasets as seed inputs, generates sufficient number of new test cases by fuzzing the seed inputs based on their context-free grammar (Context-free grammar). We illustrate the usefulness of the Fuzz-CHECKLIST by showing input diversity and identifying critical failures in state-of-the-art models for NLP task. In our experiment, we show that the Fuzz-CHECKLIST generates more test cases with higher diversity, and finds more bugs.

ACM Reference Format:

Anonymous Author(s). 2022. Programming Language Representation with Semantic-level Structure. In *Proceedings of ACM SIGSOFT International Symposium on Software Testing and Analysis (ISSTA 2022).* ACM, New York, NY, USA, 2 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

Software testing is the cruicial process when developing software. It evaluates an attribute or capability of the software and determines that it meets the requirements by examining the behavior of the software under test. Software testing in the early stage of the development finds bugs, and fixing them saves amount of costs. In addition, reliable software testing methodology ensures software quality to users in that the software meets requirements by verification and validation. Regarding that, NLP application is a branch of artifical intelligence software, and testing NLP application also becomes important process as well.

The prevalent models of NLP are evaluated via train-validation-test splits. train and validation set is used to train the NLP model and the hold-out set is used for testing by measuring accuracy. The accuracy is a indicator of the performance of the models.

Despite its usefulness, the main limitation of the testing paradaigm is that the hold-out set often overestimates the performances. Each dataset comes with specific biases, and the biases increase the discrepancy of distribution between dataset and real-world [4]. The aforementioned accuracy on hold-out set does not consider the discrepancy and it is limited to achieve comprehensive performrance of the NLP model. As a consequence, it is difficult to analyze where the errors comes from [12].

On the subject of the limitation of traditional testing paradaigm, a number of methods have been proposed. First, multiple diagnostic datasets for evaluating NLP model were introduced for obtaning generalized evaluation of the NLP model [11]. Not only that, model is evaluated on different aspects such as robustness of the model on adversarial sets [1, 2, 7, 10], fairness [3, 9], logical consistancy [5], prediction interpretations [6] and interactive error analysis [12]. Especially, CHECKLIST implements behavioral testing methodolgy for evaluating multiple linguistic capabilities of NLP model [8]. CHECKLIST introduces input-output behaviors of linguistic capabilities and generates behavior-guided inputs for validating the behaviors. It provides comprehensive behavioral testing of NLP models through a number of generated inputs. However, the approach only relies on manually generated input templates, thus the template generation becomes expensive and time consuming. In addition, the generated templates are selective and often too simple, and it is limited to provide restricted evaluation of linguistic capabilities. Thus, it does not garauntee the comprehensive evaluation.

In this paper, we propose Fuzz-CHECKLIST, a new automatic NLP model evaluation method for comprehensive behavioral testing of NLP models on sentiment analysis task. For each behavior of linguistic capability, Fuzz-CHECKLIST does not rely on the manual input generation. Instead, it establishes input requirement for evaluating a linguistic capability and finds suitable inputs that meet the requirement from existing dataset. Therefore, Fuzz-CHECKLIST increases input diversity and generalization. Further, Fuzz-CHECKLIST applies the fuzzing testing principle to generate inputs by mutating the selected inputs as seed inputs. To hold structural naturalness of mutated inputs, Fuzz-CHECKLIST analyzes context-free grammar (CFG) of the seed inputs and expand determines what CFG rules in the seed inputs to be expanded. Also, to hold contextual naturalness of the mutated inputs, Fuzz-CHECKLIST mutates both the seed and mutated inputs by replacing sentiment-independent words with context-aware suggestions.

2 BACKGROUND

3 RELATED WORK

4 APPROACH

5 RESEARCH QUESTIONS FOR EVALUATION

ISSTA 2022, 18-22 July, 2022, Daejeon, South Korea

ACM ISBN 978-x-xxxx-xxxx-x/YY/MM…$15.00

https://doi.org/10.1145/nnnnnnn.nnnnnnn

6 EXPERIMENT

7 RESULT

REFERENCES

[1] Yonatan Belinkov and Yonatan Bisk. 2017. Synthetic and Natural Noise Both Break Neural Machine Translation. *CoRR* abs/1711.02173 (2017). arXiv:1711.02173 http://arxiv.org/abs/1711.02173

[2] Mohit Iyyer, John Wieting, Kevin Gimpel, and Luke Zettlemoyer. 2018. Adversarial Example Generation with Syntactically Controlled Paraphrase Networks. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*. Association for Computational Linguistics, New Orleans, Louisiana, 1875–1885. https://doi.org/10.18653/v1/N18-1170

[3] Vinodkumar Prabhakaran, Ben Hutchinson, and Margaret Mitchell. 2019. Perturbation Sensitivity Analysis to Detect Unintended Model Biases. *CoRR* abs/1910.04210 (2019). arXiv:1910.04210 http://arxiv.org/abs/1910.04210

[4] Benjamin Recht, Rebecca Roelofs, Ludwig Schmidt, and Vaishaal Shankar. 2019. Do ImageNet Classifiers Generalize to ImageNet?. In *Proceedings of the 36th International Conference on Machine Learning (Proceedings of Machine Learning Research, Vol. 97)*, Kamalika Chaudhuri and Ruslan Salakhutdinov (Eds.). PMLR, 5389–5400. https://proceedings.mlr.press/v97/recht19a.html

[5] Marco Tulio Ribeiro, Carlos Guestrin, and Sameer Singh. 2019. Are Red Roses Red? Evaluating Consistency of Question-Answering Models. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, Florence, Italy, 6174–6184. https://doi.org/10.18653/v1/P19-1621

[6] Marco Túlio Ribeiro, Sameer Singh, and Carlos Guestrin. 2016. "Why Should I Trust You?": Explaining the Predictions of Any Classifier. *CoRR* abs/1602.04938 (2016). arXiv:1602.04938 http://arxiv.org/abs/1602.04938

[7] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. 2018. Semantically Equivalent Adversarial Rules for Debugging NLP models. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Association for Computational Linguistics, Melbourne, Australia, 856–865. https://doi.org/10.18653/v1/P18-1079

[8] Marco Tulio Ribeiro, Tongshuang Wu, Carlos Guestrin, and Sameer Singh. 2020. Beyond Accuracy: Behavioral Testing of NLP models with CheckList. In *Association for Computational Linguistics (ACL)*.

[9] Paul Röttger, Bertram Vidgen, Dong Nguyen, Zeerak Waseem, Helen Z. Margetts, and Janet B. Pierrehumbert. 2020. HateCheck: Functional Tests for Hate Speech Detection Models. *CoRR* abs/2012.15606 (2020). arXiv:2012.15606 https://arxiv.org/abs/2012.15606

[10] Barbara Rychalska, Dominika Basaj, Alicja Gosiewska, and Przemysław Biecek. 2019. Models in the Wild: On Corruption Robustness of Neural NLP Systems. In *Neural Information Processing*, Tom Gedeon, Kok Wai Wong, and Minho Lee (Eds.). Springer International Publishing, Cham, 235–247.

[11] Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R. Bowman. 2018. GLUE: A Multi-Task Benchmark and Analysis Platform for Natural Language Understanding. *CoRR* abs/1804.07461 (2018). arXiv:1804.07461 http://arxiv.org/abs/1804.07461

[12] Tongshuang Wu, Marco Tulio Ribeiro, Jeffrey Heer, and Daniel Weld. 2019. Errudite: Scalable, Reproducible, and Testable Error Analysis. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, Florence, Italy, 747–763. https://doi.org/10.18653/v1/P19-1073