TEXTFLINT: UNIFIED MULTILINGUAL ROBUSTNESS EVALUATION TOOLKIT FOR NATURAL LANGUAGE PROCESSING

Xiao Wang, Qin Liu, Tao Gui, Qi Zhang & Xuanjing Huang
Department of Computer Science
Fudan University
2005 Songhu Rd., Shanghai, China
{xiao_wang20, liuq19, tgui16, qz, xjhuang}@fudan.edu.cn

ABSTRACT

Various robustness evaluation methodologies from different perspectives have been proposed for different natural language processing (NLP) tasks, which often focus on either universal or task-specific generalization capabilities. TextFlint is a multilingual robustness evaluation toolkit for NLP tasks that incorporates universal text transformation, task-specific transformation, adversarial attack, subpopulation, and their combinations to provide comprehensive robustness analysis. It enables practitioners to automatically evaluate their models from various aspects or to customize their evaluations as desired with just a few lines of code, and generates complete analytical reports as well as targeted augmented data to address the shortcomings of the model's robustness. To guarantee user acceptability, all the text transformations are linguistically based and passed human evaluation. To validate the utility, we performed large-scale empirical evaluations (over 67,000) on state-of-the-art deep learning models, classic supervised methods, and real-world systems. TextFlint is already available at https://github.com/textflint, with all the evaluation results demonstrated at textflint.io.

1 INTRODUCTION

The detection of model robustness is attracting increasing attention in recent years, given that deep neural networks (DNNs) of high accuracy can still be vulnerable to carefully crafted adversarial examples (Li et al., 2020), distribution shift (Miller et al., 2020), data transformation (Xing et al., 2020), and shortcut learning (Geirhos et al., 2020). Existing approaches to textual robustness evaluation focus on making slight modifications to the input data, which maintains the original meaning while results in a different prediction. However, these methods often concentrate on either universal or task-specific generalization capabilities, for which it is difficult to make a comprehensive robustness evaluation.

In response to the shortcomings of recent works, we introduce TextFlint, a unified, multilingual, and analyzable robustness evaluation toolkit for NLP, which is easy to use in terms of robustness analysis. Its features include:

1. **Integrity.** TextFlint offers 20 general transformations and 60 task-specific transformations, as well as thousands of their combinations, which cover various aspects of text transformations to enable comprehensive evaluation of robustness. It also supports evaluations in multiple languages. In addition, TextFlint also incorporates adversarial attack and subpopulation (Figure 1). Based on the integrity of the text transformations, TextFlint automatically analyzes the deficiencies of a model with respect to its lexics, syntax, and semantics, or performs a customized analysis based on the needs of the user.
2. **Acceptability.** All the text transformations offered by TextFlint are linguistically based and passed human evaluation. To verify the quality of the transformed text, we conducted human

Transformation	
Original	Tasty **burgers**, and crispy fries. (Target aspect: burgers)
RevTgt	Terrible **burgers**, but crispy fries.
RevNon	Tasty **burgers**, but soggy fries.
Typos	Tatsy burgers, and cripsy fries.

Adversarial attack		
Original	Premise: Some rooms have balconies. Hypothesis: All of the rooms have balconies.	Contradiction
Adv	Premise: Many rooms have balconies. Hypothesis: All of the rooms have balconies.	Neutral

Subpopulation	
Original Set	Subpopulation - Gender
She became a nurse and worked in a hospital.	✓
I told John to come early, but he failed.	✓
The river derives from southern America.	✗
Marry would like to teach kids in the kindergarten.	✓
The storm destroyed many houses in the village.	✗

Figure 1: Examples of the three main generation functions. The example of transformation is from ABSA task, where the italic bold ***RevTgt*** (short for reverse target) denotes task-specific transformations and the bold **Typos** denotes universal transformation.

evaluation on both original and transformed texts under all of the above mentioned transformations. The transformed texts perform well in plausibility and grammaticality.

3. **Analyzability.** Based on the evaluation results, TextFlint provides a standard analysis report with respect to a model's lexics, syntax, and semantic. All the evaluation results can be displayed via visualization and tabulation to help users gain a quick and accurate grasp of the shortcomings of a model. In addition, TextFlint generates a large number of targeted data to augment the evaluated model, based on the the defects identified in the analysis report, and provides patches for the model defects.

We tested 95 the state-of-the-art models and classic systems on 6,903 transformation datasets for a total of over 67,000 evaluations, and found almost all models showed significant performance degradation, including a decline of more than 50% of BERT's prediction accuracy on tasks such as aspect-level sentiment classification, named entity recognition, and natural language inference. It means that most experimental models are almost unusable in real scenarios, and the robustness needs to be improved.

2 TEXTFLINT FRAMEWORK

According to its pipeline architecture, TextFlint can be organized into three blocks, as shown in Figure 2, (a) Input Layer, which prepares necessary information for sample generation, (b) Generation Layer, which applies data generation functions to each sample, and (c) Reporter Layer, which analyses evaluation result and generate robustness report.

2.1 INPUT LAYER

For input preparation, the original dataset, which is to be loaded by `Dataset`, should be firstly formatted as a series of JSON objects. The configuration of `TextFlint` is specified by `Config` which can be loaded from customized config file. The target model is wrapped by `FlintModel` which needs to implement specific interfaces to support specific functions.

2.2 GENERATION LAYER

Generation Layer supports three type sample generation functions to provide comprehensive robustness analysis, i.e. `Transformation`, `Subpopulation` and `AttackRecipe`. It is worth

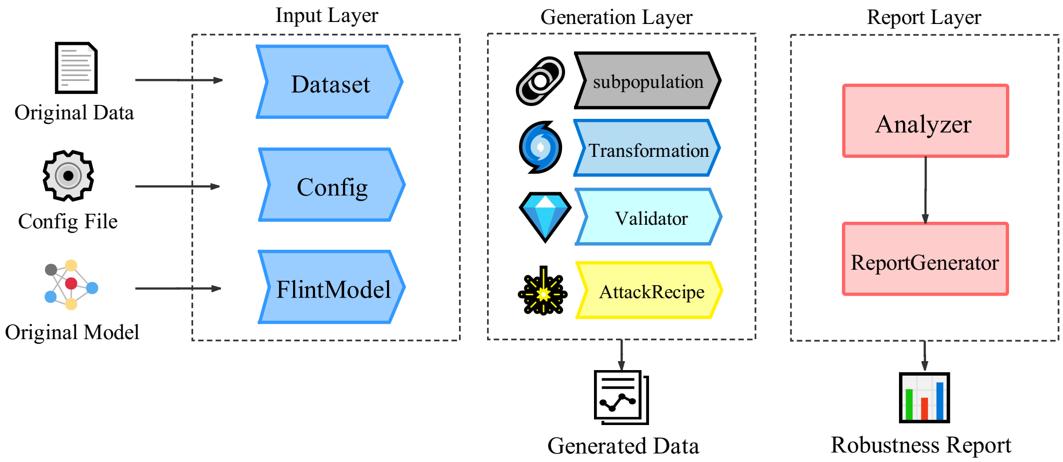

Figure 2: Architecture of TextFlint. Input Layer receives the original dataset, config file and target model as input, which are represented as `Dataset`, `Config` and `FlintModel` separately. Generation Layer consists of three parallel modules, where `Subpopulation` generates a subset of input dataset, `Transformation` augments datasets, and `AttackRecipe` interacts with the target model. Report Layer analyzes test results by `Analyzer` and provides users with robustness report by `ReportGenerator`.

noting that the procedure of `Transformation` and `Subpopulation` does not require querying the target model, which means it is a completely decoupled process with the target model prediction. Besides, to ensure semantic and grammatical correctness of transformed samples, `Validator` provides several metrics to calculates confidence of each sample.

Transformation `Transformation` aims to generate perturbations of the input text while maintaining the acceptability of transformed texts. In order to verify the robustness comprehensively, TextFlint offers 20 universal transformations and 60 task-specific transformations, as well as thousands of their combinations, covering 12 NLP tasks.

From the perspective of linguistics, the transformations are designed according to morphology, syntax, paradigmatic relation, and pragmatics. Transformations on morphology includes **KeyBoard**, **Ocr**, **Typos**, etc. As for syntactical transformations, there are **SwapSyn-WordNet**, **AddSubTree**, etc. Refer to appendix for more details.

Subpopulation `Subpopulation` is to identify the specific part of dataset on which the target model performs poorly. To retrieve a subset that meets the configuration, `Subpopulation` divides the dataset through sorting samples by certain attributes. TextFlint provides 4 general `Subpopulation` configurations, including text length, language model performance, phrase matching, and gender bias, which work for most NLP tasks.

AttackRecipe `AttackRecipe` aims to find a perturbation of an input text satisfies the attack's goal to fool the given `FlintModel`. In contrast to `Transformation` and `Subpopulation`, `AttackRecipe` requires the prediction scores of the target model. TextFlint provides 16 easy-to-use adversarial attack recipes which are implemented based on TextAttack (Morris et al., 2020).

2.3 Reporter Layer

Based on the evaluation results from Generation Layer, Report Layer provides users with a standard analysis report from syntax, morphology, pragmatics, and paradigmatic relation aspects. The running process of Report Layer can be regarded as a pipeline from `Analyzer` to `ReportGenerator`.

3 Usage

Using TextFlint to verify the robustness of specific model is as simple as running the following command:

```
$ python textflint --dataset input_file --config config.json
```

where `input_file` is the input file of csv or json format, `config.json` is a configuration file with generation and target model options.

Thanks to the design of decoupling sample generation and model verification, TextFlint can be used inside another NLP project with just a few lines of code.

```python
from textflint import Engine

data_path = 'input_file'
config = 'config.json'
engine = Engine('SA')
engine.run(data_path, config)
```

TextFlint is also avaliable for use through our web demo which is available at `https://www.textflint.com/demo`.

4 Related Work

Robustness Evaluation Many tools include evaluation methods for robustness, including NLPAug (Ma, 2019), Errudite (Wu et al., 2019), AllenNLP Interpret (Wallace et al., 2019), and Checklist (Ribeiro et al., 2020), which are only applicable to limited parts of robustness evaluations. There also exist several tools concerning robustness that are similar to our work (Morris et al., 2020; Zeng et al., 2020; Goel et al., 2021), which also include a wide range of evaluation methods. However, these tools only focus on general generalization evaluations and lack quality evaluations on generated texts or only support automatic quality constraints.

Interpretability and Error Analysis There also exist several works concerning model evaluation from different perspective. AllenNLP Interpret (Wallace et al., 2019), InterpreteML (Nori et al., 2019), LIT (Nori et al., 2019), Manifold (Zhang et al., 2018), AIX360 (Arya et al., 2019) tackles model interpretability, trying to understand the models' behavior through different evaluation methods. CrossCheck (Arendt et al., 2020), AllenNLP Interpret (Wallace et al., 2019), Errudite (Wu et al., 2019) and Manifold (Zhang et al., 2018) offer visualization and cross-model comparison for error analysis.

5 Conclusion

We introduce TextFlint, a unified multilingual robustness evaluation toolkit that incorporates universal text transformation, task-specific transformation, adversarial attack, subpopulation, and their combinations to provide comprehensive robustness analysis. TextFlint enables practitioners to evaluate their models with just a few lines of code, and then obtain complete analytical reports. We performed large-scale empirical evaluations on state-of-the-art deep learning models, classic supervised methods, and real-world systems. Almost all models showed significant performance degradation, indicating the urgency and necessity of including robustness into NLP model evaluations.

References

Dustin Arendt, Zhuanyi Huang, Prasha Shrestha, Ellyn Ayton, Maria Glenski, and Svitlana Volkova. Crosscheck: Rapid, reproducible, and interpretable model evaluation, 2020.

Vijay Arya, Rachel KE Bellamy, Pin-Yu Chen, Amit Dhurandhar, Michael Hind, Samuel C Hoffman, Stephanie Houde, Q Vera Liao, Ronny Luss, Aleksandra Mojsilović, et al. One explanation does not fit all: A toolkit and taxonomy of ai explainability techniques. *arXiv preprint arXiv:1909.03012*, 2019.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 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)*, pp. 4171–4186, 2019.

Robert Geirhos, Jörn-Henrik Jacobsen, Claudio Michaelis, Richard Zemel, Wieland Brendel, Matthias Bethge, and Felix A Wichmann. Shortcut learning in deep neural networks. *Nature Machine Intelligence*, 2(11):665–673, 2020.

Karan Goel, Nazneen Rajani, Jesse Vig, Samson Tan, Jason Wu, Stephan Zheng, Caiming Xiong, Mohit Bansal, and Christopher Ré. Robustness gym: Unifying the nlp evaluation landscape, 2021.

Linyang Li, Ruotian Ma, Qipeng Guo, Xiangyang Xue, and Xipeng Qiu. Bert-attack: Adversarial attack against bert using bert. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 6193–6202, 2020.

Edward Ma. Nlp augmentation. https://github.com/makcedward/nlpaug, 2019.

John Miller, Karl Krauth, Benjamin Recht, and Ludwig Schmidt. The effect of natural distribution shift on question answering models. In *International Conference on Machine Learning*, pp. 6905–6916. PMLR, 2020.

John X Morris, Eli Lifland, Jin Yong Yoo, and Yanjun Qi. Textattack: A framework for adversarial attacks in natural language processing. *arXiv preprint arXiv:2005.05909*, 2020.

Harsha Nori, Samuel Jenkins, Paul Koch, and Rich Caruana. Interpretml: A unified framework for machine learning interpretability, 2019.

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*, pp. 4902–4912, Online, July 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.442. URL `https://www.aclweb.org/anthology/2020.acl-main.442`.

Erik F. Tjong Kim Sang and Fien De Meulder. Introduction to the CoNLL-2003 shared task: Language-independent named entity recognition. In *Proceedings of the Seventh Conference on Natural Language Learning at HLT-NAACL 2003*, pp. 142–147, 2003. URL `https://www.aclweb.org/anthology/W03-0419`.

Eric Wallace, Jens Tuyls, Junlin Wang, Sanjay Subramanian, Matt Gardner, and Sameer Singh. AllenNLP interpret: A framework for explaining predictions of NLP models. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP): System Demonstrations*, pp. 7–12, Hong Kong, China, November 2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-3002. URL `https://www.aclweb.org/anthology/D19-3002`.

Tongshuang Wu, Marco Tulio Ribeiro, Jeffrey Heer, and Daniel S Weld. Errudite: Scalable, reproducible, and testable error analysis. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pp. 747–763, 2019.

Xiaoyu Xing, Zhijing Jin, Di Jin, Bingning Wang, Qi Zhang, and Xuan-Jing Huang. Tasty burgers, soggy fries: Probing aspect robustness in aspect-based sentiment analysis. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 3594–3605, 2020.

Guoyang Zeng, Fanchao Qi, Qianrui Zhou, Tingji Zhang, Bairu Hou, Yuan Zang, Zhiyuan Liu, and Maosong Sun. Openattack: An open-source textual adversarial attack toolkit. *arXiv preprint arXiv:2009.09191*, 2020.

Jiawei Zhang, Yang Wang, Piero Molino, Lezhi Li, and David S Ebert. Manifold: A model-agnostic framework for interpretation and diagnosis of machine learning models. *IEEE transactions on visualization and computer graphics*, 25(1):364–373, 2018.

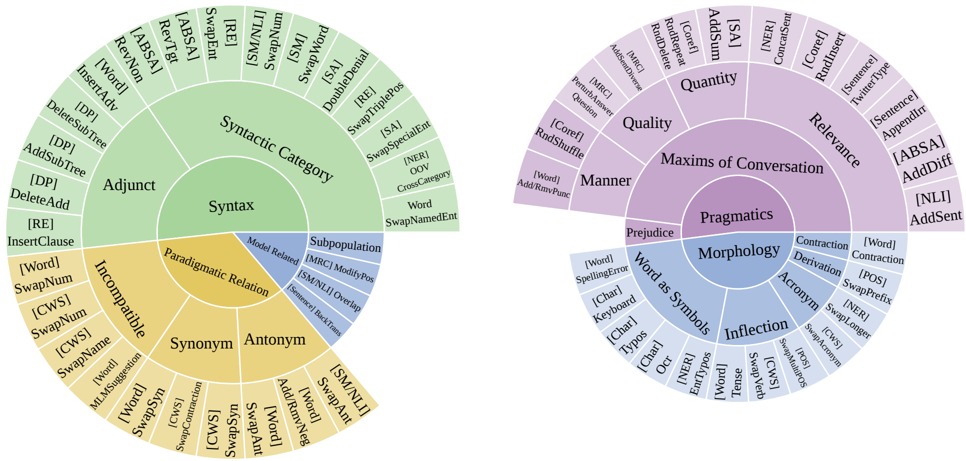

Figure 3: Overview of transformations through the lens of linguistics.

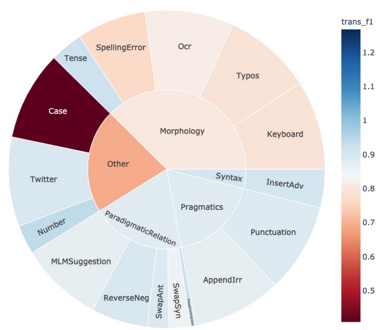

Figure 4: Robustness report of BERT base model on CONLL2003 dataset, trans_f1 represents the F1 score of the model on the transformed dataset.

A Appendix

A.1 Linguistically based Transformations

In order to verify model robustness comprehensively, TextFlint offers 20 universal transformations and 60 task-specific transformations, covering 12 NLP tasks, which are designed with respect to linguistics (Figure 3).

A.2 Robustness Report

TextFlint supports generating massive and comprehensive transformed samples within one command. By default, TextFlint performs all single transformations on original dataset to form corresponding transformed datasets, and the performance of target models is tested on these datasets. The evaluation report provides a comparative view of model performance on datasets before and after certain types of transformation, which supports model weakness analysis and guides particular improvement. For example, take BERT baseDevlin et al. (2019) as the target model to verify its robustness on CONLL2003 datasetTjong Kim Sang & De Meulder (2003), its robustness report is shown in Figure 4.