TextFlint: Unified Multilingual Robustness Evaluation Toolkit for Natural Language Processing

Xiao Wang*, Qin Liu*, Tao Gui†, Qi Zhang†, Yicheng Zou, Xin Zhou, Jiacheng Ye, Yongxin Zhang, Rui Zheng, Zexiong Pang, Qinzhuo Wu, Zhengyan Li, Chong Zhang, Ruotian Ma, Zichu Fei, Ruijian Cai, Jun Zhao, Xingwu Hu, Zhiheng Yan, Yiding Tan, Yuan Hu, Qiyuan Bian, Zhihua Liu, Shan Qin, Bolin Zhu, Xiaoyu Xing, Jinlan Fu, Yue Zhang, Minlong Peng, Xiaoqing Zheng, Yaqian Zhou, Zhongyu Wei, Xipeng Qiu and Xuanjing Huang

School of Computer Science, Fudan Univerisity

{xiao_wang20, liuq19, tgui16, qz}@fudan.edu.cn

Abstract

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 analyses. This enables practitioners to automatically evaluate their models from various aspects or to customize their evaluations as desired with just a few lines of code. TextFlint also generates complete analytical reports as well as targeted augmented data to address the shortcomings of the model in terms of its robustness. To guarantee acceptability, all the text transformations are linguistically based and all the transformed data selected (up to 100,000 texts) scored highly under 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. The toolkit is already available at https://github.com/textflint, with all the evaluation results demonstrated at textflint.io.

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 three main generation functions. The transformation example is from ABSA (Aspect-based Sentiment Analysis) task, where the italic bold ***RevTgt*** (short for reverse target) denotes task-specific transformations, and the bold **Typos** denotes universal transformation.

1 Introduction

The detection of model robustness has been 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 slightly modifying the input data, which maintains the original meaning and results in a different prediction. However, these methods often concentrate on either universal or task-specific generalization capabilities, which is difficult to comprehensively evaluate.

In response to the shortcomings of recent works, we introduce TextFlint, a unified, multilingual, and analyzable robustness evaluation toolkit for NLP. Its features include:

1. **Integrity.** TextFlint offers 20 general transformations and 60 task-specific transformations, as well as thousands of their combinations that cover a variety of aspects of text transformations to enable a comprehensive evaluation of robustness. It also supports evaluations on both English and Chinese. In addition, the toolkit also incorporates adversarial attack and subpopulation (Figure 1). Currently, 12 NLP tasks are available and more are on the way.

[*] Xiao Wang and Qin Liu contributed equally to this work and are co-first authors.

[†] Corresponding Author

Proceedings of the Joint Conference of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing: System Demonstrations, pages 347–355, August 1st - August 6th, 2021.

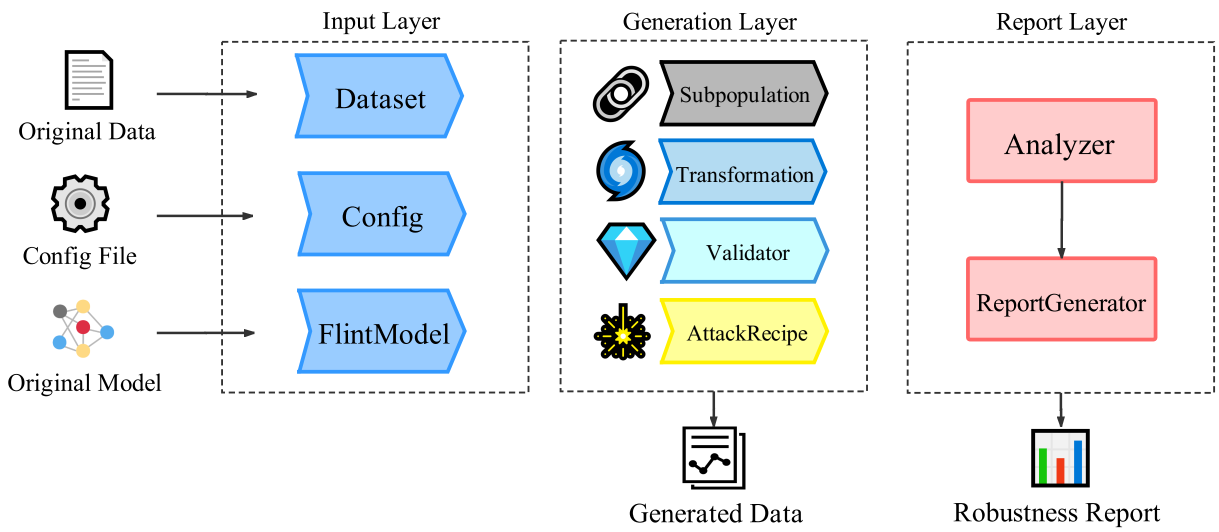

Figure 2: Architecture of TextFlint. Input Layer receives the original dataset, config file and target model as input, which are represented as `Dataset`, `Config` and `FlintModel` separately. Generation Layer consists of three parallel modules, where `Subpopulation` generates a subset of input dataset, `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`.

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 evaluation on the original and transformed texts under all of the mentioned transformations. The transformed texts performed 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 semantics. 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 amount 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 evaluated 95 state-of-the-art models and classic systems on 6,903 transformation datasets for a total of over 67,000 evaluations and found that 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. This means that the robustness of most models needs to be improved.

2 TextFlint Framework

TextFlint is designed to be flexible enough to allow practitioners to configure the workflow while providing appropriate abstractions to alleviate the concerns of the low-level implementation. According to its pipeline architecture, TextFlint can be organized into three blocks, as shown in Figure 2: (a) Input Layer, which prepares the necessary information for sample generation; (b) Generation Layer, which applies generation functions to each sample; and (c) Reporter Layer, which analyzes the evaluation results and generates a robustness report.

2.1 Input Layer

For input preparation, the original dataset, which is to be loaded by `Dataset`, should first be formatted as a series of JSON objects. The configuration of `TextFlint` is specified by `Config`, which can be loaded from a customized config file. TextFlint is model-agnostic and provides `FlintModel` to wrap the target model. This means that it can apply robustness evaluation to models implemented in any deep learning framework. After Input Layer completes the required input loading, the interaction between the system and the user is complete.

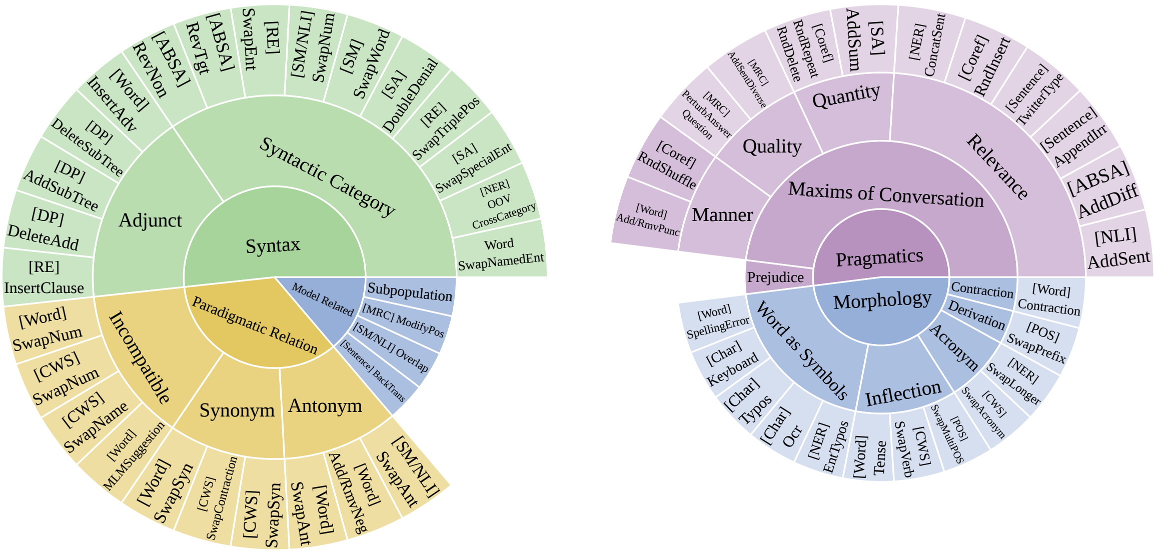

Figure 3: Overview of transformations through the lens of linguistics.

2.2 Generation Layer

Generation Layer supports three types of sample generation functions to provide comprehensive robustness analyses, i.e., `Transformation`, `Subpopulation`, and `AttackRecipe`. It is worth 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. Additionally, to ensure semantic and grammatical correctness of the transformed samples, `Validator` provides several metrics to calculate the confidence of each sample.

Transformation `Transformation` aims to generate perturbations of the input text while maintaining the acceptability of the transformed texts. 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 include **KeyBoard**, **Ocr**, **Typos**, etc. As for syntactical transformations, there are **SwapSyn-WordNet**, **AddSubTree**, etc. Due to limited space, refer to Figure 3 for specific information. Further, we conducted a large scale human evaluation on the original and transformed texts under all of the mentioned transformations (Section 4).

Subpopulation `Subpopulation` identifies the specific part of the dataset on which the target model performs poorly. To retrieve a subset that meets the configuration, `Subpopulation` divides the dataset by sorting samples according to certain attributes. TextFlint provides four general `Subpopulation` configurations, which contain **GenderBias**, **TextLength**, **LanguageModelPerplexity**, and **PhraseMatching**. Take the configuration of text length for example, **TextLength** retrieves the subset of the top 20% or bottom 20% in length.

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

Validator It is crucial to verify the quality of the samples generated by `Transformation` and `AttackRecipe`. TextFlint provides several metrics to evaluate the quality of the generated text, including (1) language model perplexity calculated based on the GPT2 model (Radford et al., 2019), (2) word replacement ratio in generated text compared with its original text, (3) edit distance between original text and generated text, (4) semantic similarity calculated based on Universal Sentence Encoder (Cer et al., 2018), and (5) BLEU score (Papineni et al., 2002).

2.3 Reporter Layer

Generation Layer yields three types of adversarial samples and verifies the robustness of the target model. Based on the evaluation results from Generation Layer, Report Layer aims to provide 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 a specific model is as simple as running the following command:

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

where `input_file` is the input file of csv or json format, and `config.json` is a configuration file with generation and target model options. Complex functions can be implemented by a simple modification on `config.json`, such as executing the pipeline of transformations and assigning the parameters of each transformation method. Take the configuration for TextCNN (Kim, 2014) model on SA (sentiment analysis) task as an example:

```
{
  "task": "SA",
  "out_dir": "./DATA/",
  "flint_model": "./textcnn_model.py",
  "trans_methods": [
    "Ocr",
    ["InsertAdv", "SwapNamedEnt"],
    ...
  ],
  "trans_config": {
    "Ocr": {"trans_p": 0.3},
    ...
  },
...
}
```

- `task` is the name of the target task. TextFlint supports 12 types of tasks.
- `out_dir` is the directory where each of the generated samples and their corresponding original samples are saved.
- `flint_model` is the python file path that saves the instance of FlintModel.

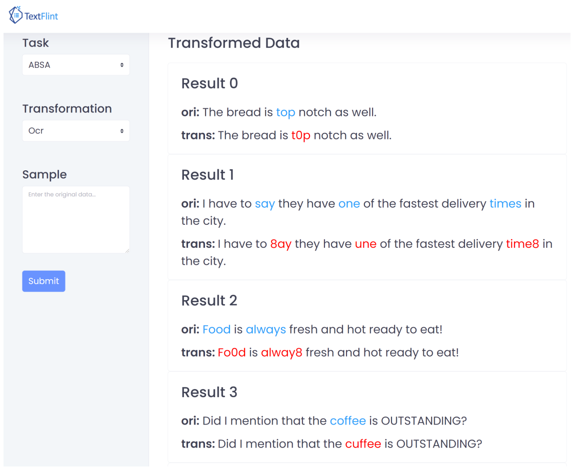

Figure 4: Screenshot of TextFlint's web interface running Ocr transformation for ABSA task.

- `trans_methods` is used to specify the transformation method. For example, `"Ocr"` denotes the universal transformation **Ocr**, and `["InsertAdv", "SwapNamedEnt"]` denotes a pipeline of task-specific transformations, namely ***InsertAdv*** and ***SwapNamedEnt***.
- `trans_config` configures the parameters for the transformation methods. The default parameter is also a good choice.

Moreover, it also supports the configuration of subpopulation and adversarial attack. For more details about parameter configuration, please move to https://github.com/textflint/textflint.

Based on the design of the 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()
engine.run(data_path, config)
```

TextFlint is also available for use through our web demo, displayed in Figure 4, which is available at https://www.textflint.io/demo.

Case Studies of Usage Due its user-friendly design philosophy, TextFlint shows its practicality in real applications. We summarize three occasions in which model robustness evaluation is challenging:

Case 1: General Evaluation For users who want to evaluate the robustness of NLP models

	Plausibility		Grammaticality			Plausibility		Grammaticality	
	Ori.	Trans.	Ori.	Trans.		Ori.	Trans.	Ori.	Trans.
DoubleDenial	3.26	3.37	3.59	3.49	***OOV***	3.69	3.76	3.54	3.48
AddSum-Person	3.39	3.32	3.76	3.59	***SwapLonger***	3.73	3.66	3.77	3.54
AddSum-Movie	3.26	3.34	3.61	3.58	***EntTypos***	3.57	3.5	3.59	3.54
SwapSpecialEnt-Person	3.37	3.14	3.75	3.73	***CrossCategory***	3.48	3.44	3.41	3.32
SwapSpecialEnt-Movie	3.17	3.28	3.70	3.49	***ConcatSent***	4.14	3.54	3.84	3.81

Table 1: Human evaluation results for task-specific transformation. Ori and Trans denote the original text and the transformed text, respectively. The table on the left is the performance of task-specific transformations for the sentiment analysis task, and the right is of that for named entity recognition. These metrics are rated on a 1-5 scale (5 denotes the best).

in a general way, TextFlint supports generating massive and comprehensive transformed samples with just one command. By default, TextFlint performs all single transformations on the original dataset to form the corresponding transformed datasets, and the performance of the 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 transformations, which supports model weakness analyses and guides particular improvements. For example, take BERT base(Devlin et al., 2019) as the target model to verify its robustness on the CONLL2003 dataset(Tjong Kim Sang and De Meulder, 2003), whose robustness report is shown in Figure 5. The performance of BERT base decreases significantly in some morphology transformations, such as ***OCR***, ***Keyboard***, ***Typos***, and ***Spelling Error***. To combat these errors of input texts and improve the robustness of the model, we suggest that placing a word correction model(Pruthi et al., 2019) before BERT would be beneficial.

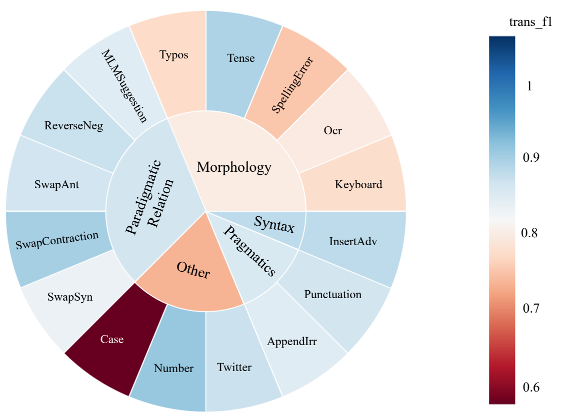

Figure 5: Robustness report of BERT base model on CONLL2003 dataset, where trans_f1 denotes the F1-score of target model on the transformed test data.

Case 2: Customized Evaluation For users who want to test model performance on specific aspects, they demand a customized transformed dataset of certain transformations or their combinations. In TextFlint, this could be achieved by modifying `Config`, which determines the configuration of TextFlint in generation. `Config` specifies the transformations being performed on the given dataset. It can be modified manually or generated automatically. By modifying the configuration, users could decide to generate multiple transformed samples on each original data sample, validate samples by semantics, preprocess samples with certain processors, and more.

Case 3: Target Model Improvement For users who want to improve the robustness of target models, they may work hard to inspect the weakness of a model with less alternative support. To tackle the issue, we believe a diagnostic report revealing the influence of comprehensive aspects on model performance will provide concrete suggestions on model improvement. By using TextFlint and applying a transformed dataset to target models, the transformations corresponding to significant performance decline in the evaluation report will provide guidance for improvements to the target models. Moreover, TextFlint supports adversarial training on target models with a large-scale transformed dataset, and the change of performance will also be reported to display performance gain due to adversarial training.

4 Benchmarking Existing Models with TextFlint

To verify the quality of transformation, we conducted human evaluation on the original and transformed texts under all of the mentioned transformations. Specifically, we considered two metrics in human evaluation: plausibility and gram-

Model	***RevTgt*** (Ori. → Trans.) Accuracy	Macro-F1	***RevNon*** (Ori. → Trans.) Accuracy	Macro-F1	***AddDiff*** (Ori. → Trans.) Accuracy	Macro-F1
Restaurant Dataset						
LSTM (Hochreiter et al., 1997)	84.42 → 19.30	55.75 → 19.88	85.91 → 73.42	55.02 → 44.69	84.42 → 44.63	55.75 → 33.24
TD-LSTM (Tang et al., 2016a)	86.42 → 22.42	61.92 → 22.28	87.29 → 79.58	60.70 → 53.35	84.42 → 81.35	61.92 → 55.69
ATAE-LSTM (Wang et al., 2016)	85.60 → 28.90	67.02 → 23.84	86.60 → 60.74	65.41 → 41.46	85.60 → 44.39	67.02 → 36.40
MemNet (Tang et al., 2016b)	81.46 → 19.30	54.57 → 17.77	83.68 → 72.95	55.39 → 45.14	81.46 → 63.62	54.57 → 39.36
IAN (Ma et al., 2017)	83.83 → 17.71	58.91 → 18.12	84.88 → 73.06	56.91 → 45.87	83.83 → 56.61	58.91 → 37.08
TNet (Li et al., 2018)	87.37 → 24.58	66.29 → 25.00	87.86 → 75.00	66.15 → 49.09	87.37 → 80.56	66.29 → 59.68
MGAN (Fan et al., 2018)	88.15 → 26.10	69.98 → 23.65	89.06 → 71.95	68.90 → 50.24	88.15 → 70.21	69.98 → 51.71
BERT-base (Devlin et al., 2019)	90.44 → 37.17	70.66 → 30.38	90.55 → 52.46	71.45 → 32.47	90.44 → 55.96	70.66 → 37.00
BERT+aspect (Devlin et al., 2019)	90.32 → 62.59	76.91 → 44.83	91.41 → 57.04	77.53 → 44.43	90.32 → 81.58	76.91 → 71.01
LCF-BERT (Zeng et al., 2019)	90.32 → 53.48	76.56 → 39.52	90.55 → 61.09	75.18 → 44.87	90.32 → 86.78	76.56 → 73.71
Average	86.83 → 31.16	65.86 → 26.63	87.78 → 67.73	64.96 → 45.15	86.83 → 66.55	65.86 → 49.49

Table 2: Accuracy and F1 score on the SemEval 2014 Restaurant dataset.

maticality[1]. For each of the transformed texts, three native speakers from Amazon Mechanical Turk were invited for evaluation, and the average score was recorded. For each kind of transformations (single one or a combination of two or more), 100 original-transformed text pairs were randomly selected for human evaluation. All of the 100,000 texts scored highly in terms of the two metrics. It was verified that the plausibility and grammaticality of the transformed texts, taking the data of SA and NER for example (Table 1), only dropped slightly compared with the original ones. Statistically, the average score of the grammaticality of the texts before and after transformation reported 3.947 and 3.825, respectively; the average of plausibility of original and transformed texts was 3.847 and 3.792, respectively. For the worst case where the grammaticality dropped the most, a decline of 1.03 from the original to the transformed text was from **Ocr** on ABSA task. The largest decline of plausibility, 0.48, was seen on the ***SwapSyn*** of CWS task.

We adopted TextFlint to evaluate hundreds of models of 12 tasks (including both English and Chinese tasks), covering various model frameworks and learning schemas, ranging from traditional feature-based machine learning approaches to state-of-the-art neural networks. All evaluated models and their implementations are available publicly. After model implementation, dataset transformation, and batch inspection, users will receive evaluation reports on various aspects, comprehensively analyzing the robustness of a system by acquiring larger test samples. From the evaluation reports, we can easily compare the model results of the original test set with those of the transformed set, spotting the main defects of the input model and identifying the model that performs the best or worst.

From the numerous evaluations and comparisons conducted by TextFlint, we have a thorough view of existing NLP systems and discovered underlying patterns about model robustness. As for the ABSA task (Table 2), methods equipped with pre-training LMs showed better performance than other models on the task-specific transformations, e.g., ***AddDiff***, where the accuracy score of BERT-Aspect dropped from 90.32 to merely 81.58. All the evaluation results and comprehensive robustness analysis are available at textflint.io.

5 Related Work

Our work is related to many existing open-source tools and works in different areas.

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, while TextFlint supports comprehensive evaluation methods, e.g., subpopulation, adversarial attacks, transformations, and so on. Besides the common transformation methods like synonym substitution and typos, various task-specific transformations are tailored for each of the 12 NLP tasks, which is peculiar to TextFlint. Moreover, we are the first to provide linguistic support for the transformations, the designs for which were inspired by linguistics and have been proved plausible and readable by human annotators.

Several tools also exist concerning robustness, which are similar to our work (Morris et al., 2020; Zeng et al., 2020; Goel et al., 2021) and include a

[1]The detailed scoring criteria are available at our website: `textflint.io`.

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. TextFlint supports both general and task-specific evaluations and guarantees the acceptability of each transformation method with human evaluations. In addition, TextFlint provides a standard report that can be displayed with visualization and tabulation. More importantly, all of the tools and modules are encapsulated within a unified framework, which completely differs from Robustness Gym (Goel et al., 2021), a simple aggregation of APIs of various tools including Checklist (Ribeiro et al., 2020) and Textattack (Morris et al., 2020). In addition, all of the transformations can be realized automatically by a simple modification to the configuration in TextFlint, while manually defined patterns are required by some of the functions in Robustness Gym.

Interpretability and Error Analysis Several works concern model evaluation from different perspectives. AllenNLP Interpret (Wallace et al., 2019), InterpreteML (Nori et al., 2019), LIT (Nori et al., 2019), Manifold (Zhang et al., 2018), and AIX360 (Arya et al., 2019) care about model interpretability in an attempt 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. TextFlint is differently motivated yet complementary with these works, which can provide comprehensive analyses on the models' defects, thus contributing to better model understanding.

6 Conclusion

We introduced 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 analyses. TextFlint enables practitioners to evaluate their models with just a few lines of code and then obtain complete analytical reports. Additionally, we also performed large-scale empirical evaluations on state-of-the-art deep learning models, classic supervised methods, and real-world systems, with all the experimental results reported on our website. Almost all models showed significant performance degradation, indicating the urgency and necessity of including robustness in NLP model evaluations.

Ethical Considerations

In consideration of ethical concerns, we provide the following detailed description:

(1) All of the transformed data comes from existing datasets, which are derived from public scientific papers. Due to the limited space, we detailed the characteristics of the dataset and the transformation methods in the README.md file at https://github.com/textflint/textflint.

(2) The quality of the transformed datasets will affect the credibility of the robustness evaluation. Compared with previous works, we additionally evaluated 100,000 samples from all of the transformation methods with respect to their plausibility and grammaticality by human evaluation.

(3) TextFlint is a robustness evaluation toolkit that does not provide any NLP models for specific tasks, such as automated essay scoring, hate speech, and so on. Therefore, there is no potential harm to vulnerable populations.

(4) Our work does not contain identity characteristics. It does not harm anyone.

(5) The subpopulation and transformation modules are executed on the CPU and do not consume a lot of computing resources. The AttackRecipe module is implemented based on TextAttack (Morris et al., 2020), which is a widely used framework for adversarial attacks and does not cause excessive computational cost.

Acknowledgements

The authors wish to thank the anonymous reviewers for their helpful comments. This work was partially funded by China National Key R&D Program (No. 2017YFB1002104), National Natural Science Foundation of China (No. 61976056, 62076069).

References

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

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. 2019. One explanation does not fit all: A toolkit and taxonomy of ai explainability techniques. *arXiv preprint arXiv:1909.03012*.

Daniel Cer, Yinfei Yang, Sheng-yi Kong, Nan Hua, Nicole Limtiaco, Rhomni St. John, Noah Constant, Mario Guajardo-Cespedes, Steve Yuan, Chris Tar, Brian Strope, and Ray Kurzweil. 2018. Universal sentence encoder for English. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 169–174, Brussels, Belgium. Association for Computational Linguistics.

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

Feifan Fan, Yansong Feng, and Dongyan Zhao. 2018. Multi-grained attention network for aspect-level sentiment classification. In *Proceedings of the 2018 conference on empirical methods in natural language processing*, pages 3433–3442.

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

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

Sepp Hochreiter, Jürgen Schmidhuber, et al. 1997. Long short-term memory. *Neural computation*, 9(8):1735–1780.

Yoon Kim. 2014. Convolutional neural networks for sentence classification. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1746–1751, Doha, Qatar. Association for Computational Linguistics.

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

Xin Li, Lidong Bing, Wai Lam, and Bei Shi. 2018. Transformation networks for target-oriented sentiment classification. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 946–956.

Dehong Ma, Sujian Li, Xiaodong Zhang, and Houfeng Wang. 2017. Interactive attention networks for aspect-level sentiment classification. In *Proceedings of the 26th International Joint Conference on Artificial Intelligence*, pages 4068–4074.

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

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

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

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

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.

Danish Pruthi, Bhuwan Dhingra, and Zachary C. Lipton. 2019. Combating adversarial misspellings with robust word recognition. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5582–5591, Florence, Italy. Association for Computational Linguistics.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.

Marco Tulio Ribeiro, Tongshuang Wu, Carlos Guestrin, and Sameer Singh. 2020. 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, Online. Association for Computational Linguistics.

Duyu Tang, Bing Qin, Xiaocheng Feng, and Ting Liu. 2016a. Effective lstms for target-dependent sentiment classification. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 3298–3307.

Duyu Tang, Bing Qin, and Ting Liu. 2016b. Aspect level sentiment classification with deep memory network. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 214–224.

Erik F. Tjong Kim Sang and Fien De Meulder. 2003. 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*, pages 142–147.

Eric Wallace, Jens Tuyls, Junlin Wang, Sanjay Subramanian, Matt Gardner, and Sameer Singh. 2019. 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*, pages 7–12, Hong Kong, China. Association for Computational Linguistics.

Yequan Wang, Minlie Huang, Xiaoyan Zhu, and Li Zhao. 2016. Attention-based lstm for aspect-level sentiment classification. In *Proceedings of the 2016 conference on empirical methods in natural language processing*, pages 606–615.

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

Xiaoyu Xing, Zhijing Jin, Di Jin, Bingning Wang, Qi Zhang, and Xuan-Jing Huang. 2020. 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)*, pages 3594–3605.

Biqing Zeng, Heng Yang, Ruyang Xu, Wu Zhou, and Xuli Han. 2019. Lcf: A local context focus mechanism for aspect-based sentiment classification. *Applied Sciences*, 9(16):3389.

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

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