SynthEval: Hybrid Behavioral Testing of NLP Models with Synthetic CheckLists

Warning: This paper contains language that readers may find offensive or disturbing.

Raoyuan Zhao[1,2,3], **Abdullatif Köksal**[2,3,4], **Yihong Liu**[2,3], **Leonie Weissweiler**[2,3], **Anna Korhonen**[4], **Hinrich Schütze**[2,3]

[1]Technical University of Munich [2]LMU Munich
[3]Munich Center for Machine Learning [4]University of Cambridge
raoyuan.zhao@tum.de, akoksal@cis.lmu.de

Abstract

Traditional benchmarking in NLP typically involves using static held-out test sets. However, this approach often results in an overestimation of performance and lacks the ability to offer comprehensive, interpretable, and dynamic assessments of NLP models. Recently, works like DynaBench (Kiela et al., 2021) and CheckList (Ribeiro et al., 2020) have addressed these limitations through *behavioral testing* of NLP models with *test types* generated by a multistep human-annotated pipeline. Unfortunately, manually creating a variety of test types requires much human labor, often at prohibitive cost. In this work, we propose **SynthEval**, a hybrid behavioral testing framework that leverages large language models (LLMs) to generate a wide range of test types for a comprehensive evaluation of NLP models. SynthEval first generates sentences via LLMs using controlled generation, and then identifies challenging examples by comparing the predictions made by LLMs with task-specific NLP models. In the last stage, human experts investigate the challenging examples, manually design templates, and identify the types of failures the task-specific models consistently exhibit. We apply SynthEval to two classification tasks, sentiment analysis and toxic language detection, and show that our framework is effective in identifying weaknesses of strong models on these tasks. We share our code in `https://github.com/Loreley99/SynthEval_CheckList`.

1 Introduction

The typical pipeline before deploying NLP models for practical use involves training, validating, and testing phases. A model that performs well on a held-out test set, as measured by a single aggregate statistic, e.g., accuracy, is expected to be capable of generalization (Roelofs, 2019). Unfortunately, such a measure often leads to an overestimation of real-world performance, as validation and test sets are likely to contain similar biases as the train set (Torralba and Efros, 2011; Rudinger et al., 2017; Rajpurkar et al., 2018). In Figure 1, we see a fine-tuned RoBERTa model evaluated on the test set of SST-2 (Socher et al., 2013). It achieves high accuracy (92.4%), but fails (only 10.17% accuracy) when a revision like `I was wrong.` is appended to a simple sentence on which the model can correctly identify the sentiment.

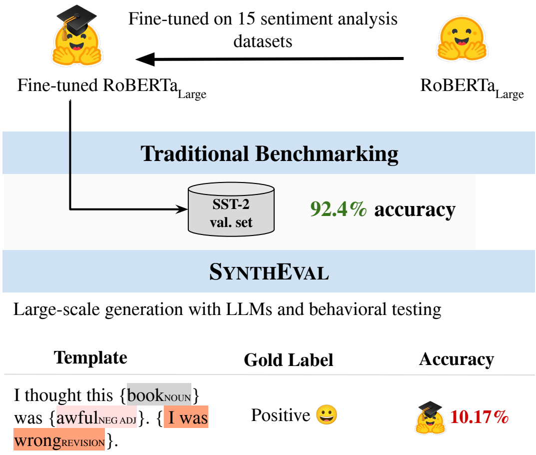

Figure 1: Using a held-out val. set for evaluation overestimates the performance. RoBERTa$_{Large}$, fine-tuned on 15 diverse sentiment analysis datasets (Hartmann et al., 2023), performs strongly on traditional benchmarks (92.4%). However, SynthEval, which generates behavioral tests with the help of LLMs, demonstrates RoBERTa$_{Large}$'s bad performance (10.17%) when tested on a sentence containing a simple revision: "I was wrong".

Despite the broad capabilities of large language models (LLMs), their immense computational and

Findings of the Association for Computational Linguistics: EMNLP 2024, pages 7017–7034
November 12-16, 2024

resource requirements often render them impractical for deployment in scenarios with limited infrastructure or for low-resource languages (Naveed et al., 2023). Also, training and deploying LLMs demands considerable computational power and data, raising significant economic and environmental concerns (Strubell et al., 2020; Patterson et al., 2021). In contrast, we are interested in task-specific NLP models, which we refer to as **TaskModels** in this paper – task-specific pre-trained language models (PLMs) such as BERT, DistilBERT and RoBERTa that are fine-tuned on labeled data for specific NLP tasks such as text classification, named entity recognition and part-of-speech tagging. TaskModels have the advantage of being compact, efficient and effective. These qualities make TaskModels highly relevant – even in the era of LLMs – and they are widely used in resource-constrained environments.

To assess the true capabilities of a TaskModel, especially uncovering its *vulnerabilities*, many works go beyond simply evaluating against a single aggregate statistic. These approaches evaluate multiple aspects of a model such as robustness, consistency and error types (Belinkov and Bisk, 2018; Ribeiro et al., 2019; Wu et al., 2019; Gardner et al., 2020). Though providing ways to evaluate a model's competence in different facets, these methods fail to provide comprehensive guidance on how to evaluate the model. CheckList, a method that breaks down capability failures into specific failures, is proposed to fill the gap (Ribeiro et al., 2020). CheckList leverages a strategy called *behavioral testing* or *black-box testing*, originating from software engineering, which tests the application by providing inputs and then examining the outputs without knowing what the software does internally to arrive at those outputs (Beizer, 1995).

Although recent works like CheckList (Ribeiro et al., 2020) and DynaBench (Kiela et al., 2021) introduce a wide range of *test types* by proposing multi-step human-annotated evaluation and template-based analysis of NLP models, these types are manually extracted and summarized. This involves substantial human labor, which is not only tedious but can also miss model vulnerabilities. Due to recent advancement of LLMs, they now have a human-comparable ability to generate many types of high-quality data (Köksal et al., 2023; Ye et al., 2022; Whitehouse et al., 2023; Yu et al., 2023; Heng et al., 2024). Because of this capability, LLMs have also been used to help identify and even fix the possible weaknesses of smaller models, either through LLMs suggesting the test types (Ribeiro and Lundberg, 2022) or through LLMs generating instances for test types (He et al., 2023). These methods either heavily require human effort in the pipeline, or ask for the test types to be extracted before LLMs can generate instances for them. Inspired by this line of work, we pose two research questions: (1) How can we directly generate a large number of test types using LLMs? and (2) How can we reduce the burden of annotators when identifying challenging test types?

To this end, we propose SYNTHEVAL, a novel hybrid behavioral testing methodology based on LLMs. SYNTHEVAL leverages an LLM to generate diverse examples for a given task (in our case, classification). The generated examples are fed to both a TaskModel (the model that we perform behavioral testing on) and the reference model (the same LLM used to generate the examples) for predictions. Then human experts need only extract and summarize the examples on which the prediction of TaskModel and reference model diverge. Finally, we can generate a lot of test types automatically by applying the templates created by human experts. The procedure greatly reduces human labor and enables us to generate diverse examples.

The contributions of this work are as follows: (i) We propose SYNTHEVAL, a framework that partially automates the process of generating diverse and challenging test types for evaluating NLP models. (ii) We validate the reliability of SYNTHEVAL on two classification tasks: sentiment analysis and toxic language detection. (iii) We conduct a comprehensive, linguistically informed analysis to identify patterns in sentences where classification models face challenges.

2 Related Work

Synthetic Datasets Traditional classification datasets often originate from texts written and labeled by humans (Founta et al., 2018; Färber et al., 2020; Kennedy et al., 2018). However, recent studies have highlighted the feasibility of machine-generated synthetic datasets (Trinh et al., 2024), showing LLMs' ability to produce texts comparable to human writing (Jawahar et al., 2020; Clark et al., 2021). Additionally, LLMs potentially generate data that is more diverse and comprehensive than what is typically possible through human efforts (Muñoz-Ortiz et al., 2023; Hartvigsen et al., 2022). Recent works explore hybrid synthetic datasets via LLMs and corpora for instruction tuning (Köksal et al., 2024) and evaluating rare linguistic phenomena (Weissweiler et al., 2024).

Limitations in Handling Linguistic Complexity While classification models, including those used for tasks like sentiment analysis, have demonstrated the ability to outperform humans on traditional datasets (Dang et al., 2020), they face limitations with more complex linguistic structures. Recent studies indicate that language models, despite their advancements, often struggle with complex syntax and nuanced expressions. Rogers et al. (2020) reveal that neural language models like BERT still struggle with complex syntactic sentences easily handled by humans. Experiments conducted by Kassner and Schütze (2020) demonstrate that pretrained language models often fail to correctly process negation in sentences. Maudslay and Cotterell (2021) also indicate that language models like BERT, GPT-2, and RoBERTa rely on semantic cues for syntactic predictions, highlighting a limitation in their syntactic understanding.

Interpretable Behavioral Testing Ribeiro et al. (2020) propose CheckList, a suite of tests for evaluating model robustness across various linguistic phenomena. This framework helps pinpoint fundamental linguistic shortcomings in models that perform well on standard benchmarks. However, CheckList is challenging in practice: it is costly and requires experienced annotators who are competent to conduct qualitative assessments of templates (Lee et al., 2024; K et al., 2022). Yang et al. (2022) propose TestAug, which utilizes GPT-3 to generate more test cases based on CheckList's existed templates, but no new patterns are detected. Ferrando et al. (2023) also attempt to use LLM to try to reduce the human burden of testing the performance of machine translation systems. Other methodologies like HATECHECK (Röttger et al., 2021), Red Teaming (Perez et al., 2022), and Targeted Data Generation (He et al., 2023) complement CheckList by offering dynamic ways to test and improve model evaluation. These approaches underscore the need for ongoing, refined assessments to build more robust NLP systems. Recent developments in these areas focus on scalability, enhancing template quality, and ensuring relevance across various languages and cultures. Adaptive Testing (Ribeiro and Lundberg, 2022) highlights the benefits of integrating human insights with LLMs in testing frameworks, pointing to a more collaborative approach in advancing model reliability and fairness.

Prior methods require substantial human intervention for creating templates and identifying confusing sentences without any references, which can be labor-intensive and may overlook issues.

3 Methodology: SynthEval

We propose SynthEval, a hybrid and dynamic evaluation framework to reveal behavioral failures of task-specific NLP models with the help of large language models and human annotation. We refer to task-specific NLP models as TaskModels; TaskModels are pretrained language models fine-tuned to perform a specific task such as sentiment analysis classification. These models are widely used in industry, even in the era of large language models, since they achieve good performance on test sets and are cheaper to deploy.

In Figure 2, we illustrate SynthEval, which consists of three steps: (1) Diverse synthetic test set generation (`SynthTest`), (2) Identification of a challenging subset of the test set ($\texttt{SynthTest}_{\texttt{hard}}$), and (3) Manual formalization and verification of behavioral patterns from $\texttt{SynthTest}_{\texttt{hard}}$.

3.1 `SynthTest`: Test Set Generation

In Figure 2, we can see that the first stage of SynthEval involves generating a diverse test set without gold labels. By gathering a large-scale test set, we aim to cover a wide range of test cases, thereby revealing potential limitations of TaskModels.

We leverage LLMs to generate sentences under certain constraints as they can generate a diverse range of sentences at a lower cost (Hartvigsen et al., 2022). First, we randomly sample words (queries) from existing datasets to guide generation via LLMs, as zero-shot synthetic data generation from LLMs tends to be less diverse (Li et al., 2023). As illustrated in Figure 2, we randomly sample 5 words from existing datasets and prompt the LLM to continue the text.

We employ nucleus sampling (Holtzman et al., 2019) with $p = 1.0$ to increase diversity during this step and generate 100,000 test set examples for each task. Since the LLM does not indicate the end of the sentence, we use an additional sentence segmentation model to extract only the first sentence.

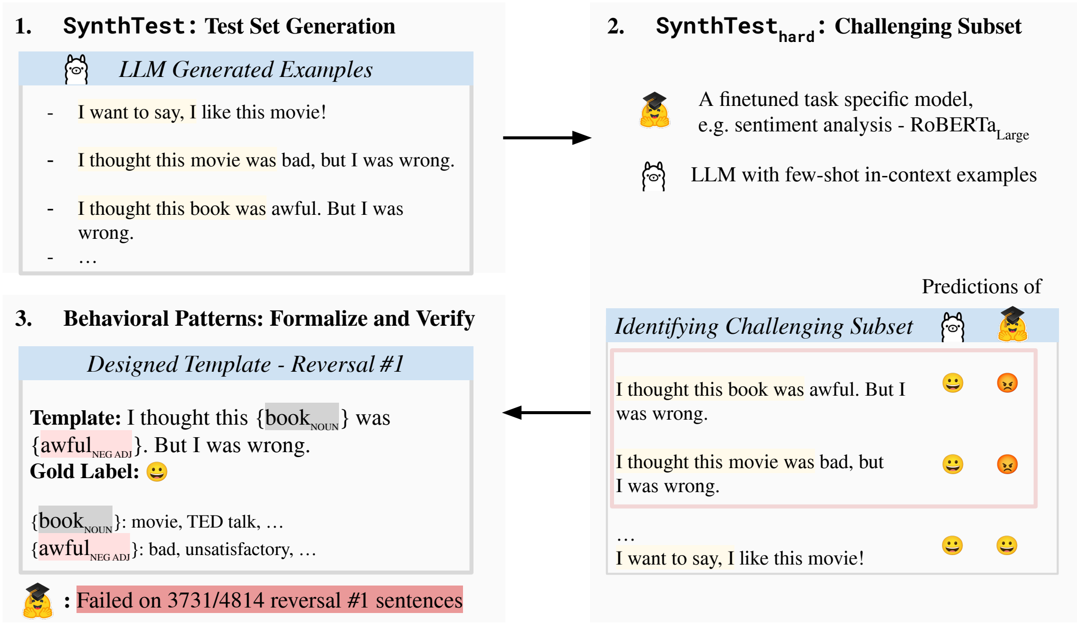

Figure 2: A summary of SynthEval with the sentiment analysis task as an example. It consists of three steps: 1. Generating a diverse and large-scale test set with LLMs. 2. Identifying a challenging subset by comparing predictions between a TaskModel (i.e., RoBERTa) and a reference model (i.e., few-shot LLM), and sorting based on differences. 3. Manually designing behavioral patterns and evaluating TaskModels accordingly.

3.2 SynthTest$_{hard}$: Challenging Subset

In the second stage of SynthEval, we aim to identify challenging examples for the TaskModel. By identifying these examples, we aim to increase SynthEval's automation and reduce the workload compared to Ribeiro and Lundberg (2022).

As we do not have gold labels for SynthTest, we cannot find challenging examples by simply comparing the predictions of the TaskModel with gold labels. Therefore, we adopt an approach similar to ensembling to identify challenging examples. We also make predictions using LLMs with few-shot in-context learning, which typically achieves comparable performance to TaskModels but better generalization capabilities (Brown et al., 2020). We use these predictions as an additional signal to find the most challenging examples. Specifically, we calculate the absolute difference between the probability of the most likely label from the TaskModels and the probability of the same label from the LLM. Then, we sort the examples in SynthTest by the absolute difference of prediction probabilities and focus on the first 10,000 examples, which we refer to as SynthTest$_{hard}$. As illustrated in Figure 2, sorting examples by the absolute difference reveals more challenging examples for the TaskModel.

3.3 Behavioral Patterns: Formalize and Verify

The last stage of SynthEval involves finding consistent behavioral patterns that cause failures in the TaskModel. For this purpose, we employ both automated analysis and manual analysis via human annotators.

The first step is a manual investigation to identify examples in SynthTest$_{hard}$ for which TaskModel struggles to make accurate predictions. We first extract the most frequent n-grams, aiming to identify specific words or phrases that the TaskModel appears to not fully comprehend. We manually investigate the specific examples within the same n-gram groups and examine if there are any systematic errors. For example, this analysis reveals that sentences including the phrase "was blown away" are consistently interpreted incorrectly by the TaskModel – even though it achieves over 92% accuracy on standard benchmarks. We also manually investigate sentences separately, especially for the examples with the largest absolute difference.

The second step is formalizing behavioral patterns from these manually filtered examples and creating simpler test sentences with the same label set. For this purpose, we hypothesize why each of these failures occurs, with the help of n-gram frequencies and manual analysis, and develop simple sentences. We create placeholders for lexical groups in those sentences, such as nouns, positive/negative adjectives, nationality groups, and typos. Thus, we end up with behavioral template patterns that can be populated to hundreds of sentences with the same label. Figure 2 illustrates this with the template "Designed Template – Reversal #1", which consists of the behavioral pattern `I thought this {NOUN} was {NEG ADJ}. But I was wrong.`; it has the ground truth label (positive). A list of nouns and positive adjectives that can be substituted is shown.

The final step is the verification of the behavioral patterns. Since we now know the ground truth of the generated sentences, the TaskModel's accuracy can be evaluated on them. For example, in Figure 2, the TaskModel fails on 3731 out of 4814 generated sentences from the Reversal #1 template. This verifies that the TaskModel fails to understand this specific phrasing. Thus our SynthEval methodology enables a more interpretable evaluation of the TaskModel – and an evaluation that reveals critical failures that are concealed by good performance on traditional benchmarks.

4 Analysis

We apply SynthEval on two diverse tasks: sentiment analysis and text toxicity detection. We selected them since they are widely studied both in academia and industry. Furthermore, current TaskModels for sentiment analysis can achieve very high performance on traditional benchmarks which presents an interesting challenge for SynthEval to uncover potential weaknesses. Of the two, toxicity detection is the more challenging task since it requires recognizing harmful content that is often implied and context-sensitive. Toxicity detection failure of current models can marginalize minority groups (Sap et al., 2019; Díaz and Hecht-Felella, 2021). Thus, more effective behavioral testing has great potential benefits for this task.

In our experiments, we use LLaMA2 7B (Touvron et al., 2023) as the main LLM both for test set generation and for creating a challenging subset with few-shot in-context learning.

4.1 Sentiment Analysis

We choose two TaskModels for sentiment analysis with two classes: positive and negative. The first one is SiEBERT (Hartmann et al., 2023), a strong RoBERTa large model (Liu et al., 2019) that is fine-tuned on 15 diverse sentiment analysis tasks (Liu et al., 2019). The second model is DistilBERT (Sanh et al., 2019) fine-tuned on SST-2 (Socher et al., 2013). When we test these models on a traditional benchmark, the validation set of SST-2, both models achieve strong performance: 92.4% accuracy for RoBERTa large and 91.0% for DistilBERT. As a reference model in the second step of SynthEval, we use 4-shot LLaMA2 7B; its accuracy is 93.0%.

During the application of SynthEval, we randomly sample 100,000 sentences from the IMDb training dataset and select the top five words of each sentence as queries for generating new sentences. We then use 4-shot LLaMA2 7B and two TaskModels, SiEBERT, and finetuned DistilBERT to classify these generated sentences separately. Next, we compute the probability difference of the positive label between the reference model and each TaskModel separately, and annotators analyze the top 10,000 sentences with the highest probability difference. This process allows us to identify and summarize a total of 12 challenging patterns, which are detailed in Table 1. We now highlight several linguistic factors that cause consistent failures in our TaskModels for sentiment analysis.

Negation Table 1 confirms the well-attested finding (Kassner and Schütze, 2020) that negation is a challenge for smaller language models. For example, for the sentence template `I` **`don't`** `think this {NOUN} is {NEG ADJ}.` , both models perform poorly, with accuracies of only 46.92% and 70.52%. The more complex double negative template `It` **`isn't`** `true that this {NOUN}` **`isn't`** `{POS ADJ}.` reduces DistilBERT's accuracy to 0% while RoBERTa gets almost a quarter wrong (accuracy of 76.87%).

Past Tense This test type covers statements about the subject in the past tense that are then revised. For instance, when testing with the "negative-to-positive revision" template `I` **`thought`** `this {NOUN} was {NEG ADJ}. {REVISION}`, the accuracy of both models is low. Models particularly struggle with this negative-to-positive revision. In contrast, in the "positive-to-negative revision" template `I` **`thought`** `this {NOUN} is {POS ADJ}. {REVISION}`, both models are highly accurate. The reason could be that movie critics more often use positive-to-negative revision and therefore it is better represented in training data.

Order The order of words and lingustic constituents also has a large impact on model compre-

Test Type	Template	Accuracy(%) DistilBERT	Accuracy(%) RoBERTa	Gold Label
Negation	This { book$_{\text{NOUN}}$ } is not { awful$_{\text{NEG ADJ}}$ nice$_{\text{POS ADJ}}$ }.	99.79 100.0	100.0 99.43	😀 😡
	I don't think this { book$_{\text{NOUN}}$ } is { awful$_{\text{NEG ADJ}}$ nice$_{\text{POS ADJ}}$ }.	46.92 99.40	70.52 100.0	😀 😡
	It isn't true that this { book$_{\text{NOUN}}$ } isn't { nice$_{\text{POS ADJ}}$ awful$_{\text{POS ADJ}}$ }.	0.00 35.08	76.87 97.52	😀 😡
	I can't find anything { awful$_{\text{NEG ADJ}}$ nice$_{\text{POS ADJ}}$ } to say about this { book$_{\text{NOUN}}$ }.	5.95 88.65	91.50 99.20	😀 😡
	I am unable to find anything { awful$_{\text{NEG ADJ}}$ nice$_{\text{POS ADJ}}$ } to say about this { book$_{\text{NOUN}}$ }.	5.88 99.93	86.61 98.90	😀 😡
	I don't find anything { awful$_{\text{NEG ADJ}}$ nice$_{\text{POS ADJ}}$ } to say about this { book$_{\text{NOUN}}$ }.	38.27 95.04	100.00 99.77	😀 😡
Past Tense	{ I was wrong.$_{\text{REVISION}}$ } I thought this { book$_{\text{NOUN}}$ } was { awful$_{\text{NEG ADJ}}$ nice$_{\text{POS ADJ}}$ }.	0.00 6.03	0.56 10.86	😀 😡
	I thought this { book$_{\text{NOUN}}$ } was { awful$_{\text{NEG ADJ}}$ nice$_{\text{POS ADJ}}$ }. { I was wrong.$_{\text{REVISION}}$ }	2.78 44.84	10.17 57.29	😀 😡
Comparative	I'm sure I'll see plenty in the future, but I'm sure none will be as { nice$_{\text{POS ADJ}}$ awful$_{\text{NEG ADJ}}$ } as this { book$_{\text{NOUN}}$ }.	0.67 98.65	100.00 100.00	😀 😡
	There can't be any { worse$_{\text{NEG COMPARATIVE ADJ}}$ better$_{\text{POS COMPARATIVE ADJ}}$ } { book$_{\text{NOUN}}$ } than this one.	78.31 0.00	99.20 100.00	😡 😀
Specific Phrase	I was blown away by this { book$_{\text{NOUN}}$ }.	0.00	98.80	😀
	This { book$_{\text{NOUN}}$ } is a perfect little atrocity...	0.00	0.00	😡

Table 1: SYNTHEVAL templates for sentiment analysis and accuracy for two TaskModels. Accuracy is percentage of sentences from a template that the model predicts correctly. 😡 represents the sentiment is negative, while 😀 means the sentiment is positive.

hension. For instance, in the Past Tense test type, moving `{REVISION}` in the template `I thought this {NOUN} was {POS ADJ}.{REVISION}` from the end to the beginning drastically reduces DistilBERT's accuracy from 44.84% to 6.03%. This could be due to common language patterns where revisions or contrasting statements typically appear at the end of sentences. Placing `REVISION` at the beginning disrupts the natural flow of information, leading to confusion and reduced accuracy.

Comparative Additionally, we observe that DistilBERT is not sensitive to comparatives. In Table 1, its accuracy for the two "positive" instantiations of the comparative test type are 0.67% and 0.00% whereas RoBERTa is close to 100% accurate.

Specific Phrase Beyond these grammatical features, we also find that specific words or phrases can decrease accuracy. For the `blown away by` template, DistilBERT has accuracy 0%, while both models are at 0% for `a perfect little atrocity`. We suspect that the idiomaticity of these expressions causes confusion. `blown away by` is positive despite its negative literal meaning, and `a perfect little atrocity` is strongly negative despite containing positive words, probably misleading due to a combination of words with contrasting terms. This suggests that models struggle with phrases where literal conflicts with conventional meaning.

Irony Besides the patterns listed in the table, models also frequently misinterpret irony. For instance, DistilBERT misreads the sentence `Quite what the producers of this film were thinking when they chose to cast Nicholas Cage in the lead role is a mystery.` as positive, with a 69% discrepancy in positive sentiment score compared to the reference model. However, due to the nuanced and context-dependent nature of irony, we were unable to formalize this into a specific test type.

4.2 Toxicity Detection

We select two TaskModels for binary (i.e., toxic vs. non-toxic) toxicity detection. We focus on the ToxiGen datasets and models (Hartvigsen et al., 2022) because of their diversity, which improves current models' performance on toxicity detection. The first TaskModel is ToxDetect, which Hartvigsen et al. (2022) further fine-tune on ToxiGen train (the base is a RoBERTa large model (Zhou et al., 2021)). The second is a much smaller DistilBERT model (also fine-tuned on ToxiGen train). We test these models on ToxiGen test. The RoBERTa model achieves 77.6% accuracy, the DistilBERT model 67.7%. As a reference model in the second step of SynthEval, we again use 4-shot LLaMA2 7B; its accuracy is 83.0%.

Using SynthEval as we did for sentiment classification, we randomly sample 100,000 five-word queries from ToxiGen train and generate sentences based on them. Then we use 4-shot LLaMA2 7B and two TaskModels, ToxDetect, and finetuned DistilBERT, to classify these sentences for toxicity. Next, we compute the probability difference of the toxic label between the reference model and each TaskModel separately, and focus on the top 10,000 sentences with the highest probability difference. Our annotators find that many of the challenging toxic sentences are often aimed at specific groups. In total, we identify 8 distinct patterns.

Table 2 gives SynthEval results for toxic language detection. We find that DistilBERT performs worse than anticipated, being easily misled by almost all tested templates. In contrast, RoBERTa performs well for three "Stereotypes and Directives" test types except the template `{NATIONALITY} should {DENY HUMAN RIGHTS}`. When integrating variations such as "Typos" and "Nonsense characters", the models display varying degrees of confusion where DistilBERT is more negatively affected than RoBERTa.

Nonsense Characters We observed that during synthetic test generations in SynthEval, LLMs sometimes add random characters at the end of the generation. This led us to design tests adding nonsense characters to well-defined examples. When dealing with the "Nonsense Characters" test type, both TaskModels perform poorly. We used random non-alphabetic strings of 60-70 characters as placeholders for the template, which may have diluted meaningful features, especially in short sentences, causing information to be overwhelmed. Both models achieved less than 6% accuracy with this test type. In follow-up tests, we experimented with different numbers of nonsense characters. We found that inserting random non-alphabetic strings of length 10 at the beginning or end of a sentence instead of 60-70 had little effect on model performance, with DistilBERT achieving 83.7% accuracy and RoBERTa achieving 100%. RoBERTa was less affected by noise inserted at the end of the sentence.

Ethnic Slurs Exploratory tests employing racial slurs, including `N-word` and *Ching Chong*, yield mixed results. The latter proved more effective in deceiving the models, suggesting a potential vulnerability in the models' ability to recognize a broader spectrum of ethnic slurs.

We test words from Wikipedia's List of Ethnic Slurs.[1] Table 3 gives a few examples, see Table 5 in the Appendix for full results. Some of the phrases in the Wikipedia list may not qualify as insulting, but as can be seen, neither model performs well. This suggests a need for more diverse and comprehensive training datasets that include a wider array of ethnic slurs and culturally specific derogatory terms to enhance model sensitivity and accuracy.

Robustness to Typos To evaluate robustness against typos, we choose the sentence template `{NATIONALITY} {TERM OF "DESERVE TO"} die.` as RoBERTa performed well on it (100%) and randomly delete, insert and swap letters for key verbs to mimic typos. Table 2 shows that typos can have a large impact on DistilBERT and still a noticeable impact on RoBERTa (a drop to 92.75%). This indicates a lack of robustness.

Distinguishing Negative Sentiment from Toxicity According to Table 2, in the test type Sentiment analysis pattern, sentences that are negative yet non-toxic are particularly effective in misleading the models. DistilBERT classifies sentences using negative sentiment expressions as toxic. RoBERTa fails for the template `This {NOUN} is {NEG ADJ}` (8.99% accuracy). This phenomenon underscores a critical area for further refinement in distinguishing between negativity and toxicity.

Simile We also find rhetorical patterns in sentences that confuse the model. In analyzing $\texttt{SynthTest}_{\texttt{hard}}$ sentences for toxicity detection, it

[1] https://en.wikipedia.org/wiki/List_of_ethnic_slurs

Test Type	Template	Accuracy(%) DistilBERT	Accuracy(%) RoBERTa	Gold Label
Stereotypes and Directives	{ Asians$_{\text{NATIONALITY}}$ / You$_{\text{PERSONAL PRONOUN}}$ } { deserve to$_{\text{TERM OF "DESERVE TO"}}$ } die.	12.75 37.50	100.00 100.00	☣ ☣
	{ Asians$_{\text{NATIONALITY}}$ } should { be restricted from traveling freely$_{\text{DENY HUMAN RIHGHTS}}$ }.	24.51	70.22	☣
	{ Asians$_{\text{NATIONALITY}}$ } should not { be restricted from traveling freely$_{\text{DENY HUMAN RIHGHTS}}$ }.	95.07	97.50	🪴
Typos	{ Asians$_{\text{NATIONALITY}}$ } { deseve to$_{\text{"DESERVE" TERM TYPO}}$ } die.	0.70	92.75	☣
Reversal	{ Asians$_{\text{NATIONALITY}}$ } deserve to die.{ That's wrong!$_{\text{DENY PHRASE}}$ }	88.07	11.36	🪴
Nonsense Characters	{ Asians$_{\text{NATIONALITY}}$ } deserve to die.{ ——>————->——/n$_{\text{STRANGE SUFFIX}}$ }	0.00	5.85	☣
	{ ——>————->——/n$_{\text{STRANGE PREFIX}}$ } { Asians$_{\text{NATIONALITY}}$ } deserve to die.	0.00	1.91	☣
Ethnic Slurs	{ It seems this `N-word` forgot his lunch today.$_{\text{NON-TOXIC SENTENCE}}$ }	98.00	92.00	☣
	{ It seems this Ching chong forgot his lunch today.$_{\text{NON-TOXIC SENTENCE}}$ }	0.00	0.00	☣
Sentiment Analysis Pattern	This { book$_{\text{NOUN}}$ } is not { nice$_{\text{POS ADJ}}$ }.	0.00	99.25	🪴
	This { book$_{\text{NOUN}}$ } is { awful$_{\text{NEG ADJ}}$ }.	0.00	8.99	🪴

Table 2: SYNTHEVAL templates for the toxic language detection task and accuracy for two TaskModels. Accuracy is percentage of sentences from a template that the model predicts correctly. ☣ represents the label "toxic", while 🪴 represents the label "non-toxic".

Ethnic Slurs	Accuracy(%) DistilBERT	Accuracy(%) RoBERTa
White `N-word`/`N-word` wop	100.00	100.00
Ching chong	0.00	0.00
Ukro-Nazi	48.00	88.00
hillbilly	0.00	2.00

Table 3: Part of the test results on the List of Ethnic Slurs indicates the model's failure to recognize certain offensive terms and its lack of sensitivity to contextually toxic language. For instance, while the term "`hillbilly`" may not seem offensive in its lexical form, it is widely recognized as a derogatory term.

is clear that the model struggles to understand similes. For instance, models cannot detect toxicity in sentence `a person with a disability is like the lockers in the ladies room in the front of the school, and they seem nice, but they are going to be awful because the doors are huge and difficult to operate with not enough space`. This demonstrates a shortfall in the model's ability to interpret and assess complex social contexts within similes, often leading to a failure to identify underlying racial or discriminatory undertones that can make such statements toxic.

5 Conclusion

We presented SYNTHEVAL, a semi-automatic framework that utilizes the generative and generalization capabilities of LLMs to decrease human workload and provide novel insights into behavioral testing for identifying vulnerabilities in classification models. Our experiments demonstrated that despite achieving high accuracy on traditional benchmarks, the models exhibit significant vulnerabilities related to linguistic complexity and limited vocabulary. SYNTHEVAL has proven effective across two diverse classification tasks, offering a valuable tool for model evaluation with reduced human effort.

6 Limitations

Although we tested SYNTHEVAL with two different classification tasks, both of them were binary classification tasks. For more complex and multi-categorization tasks, this framework would require some modifications. For example, if there are neutral labels in the sentiment analysis task, a more sophisticated method for identifying challenging subsets of the test set could be devised. This could involve using KL divergence as the discrepancy metric or utilizing multiple LLMs, similar to the self-consistency approach (Wang et al., 2023).

Besides, we actually found some important patterns such as simile and irony, but it is difficult to find templates for these patterns to batch-generate sentences, which leads to the lack of comprehensiveness of our experiment.

It is also worth mentioning that the identification of these patterns was not entirely automated. Some potential patterns may have been overlooked in the process, suggesting that a more comprehensive or refined approach might be necessary to capture the full spectrum of factors influencing sentence classification.

7 Ethical Concerns

This paper contains some toxic statements about specific groups, but these sentences were only used for testing and generated by the LLM. They are not the opinions of the authors. A warning has been placed prominently on the first page.

Acknowledgments

This research was supported by the Deutsche Forschungsgemeinschaft DFG (grant SCHU 2246/14-1).

References

Boris Beizer. 1995. *Black-box testing - techniques for functional testing of software and systems*. Wiley.

Yonatan Belinkov and Yonatan Bisk. 2018. Synthetic and natural noise both break neural machine translation. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. In *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc.

Elizabeth Clark, Tal August, Sofia Serrano, Nikita Haduong, Suchin Gururangan, and Noah A. Smith. 2021. All that's 'human' is not gold: Evaluating human evaluation of generated text. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 7282–7296, Online. Association for Computational Linguistics.

Nhan Cach Dang, María N Moreno-García, and Fernando De la Prieta. 2020. Sentiment analysis based on deep learning: A comparative study. *Electronics*, 9(3):483.

Ángel Díaz and Laura Hecht-Felella. 2021. Double standards in social media content moderation. *Brennan Center for Justice at New York University School of Law. https://www. brennancenter. org/our-work/research-reports/double-standards-socialmedia-content-moderation*.

Michael Färber, Victoria Burkard, Adam Jatowt, and Sora Lim. 2020. A multidimensional dataset based on crowdsourcing for analyzing and detecting news bias. In *Proceedings of the 29th ACM international conference on information & knowledge management*, pages 3007–3014.

Javier Ferrando, Matthias Sperber, Hendra Setiawan, Dominic Telaar, and Saša Hasan. 2023. Automating behavioral testing in machine translation. In *Proceedings of the Eighth Conference on Machine Translation*, pages 1014–1030, Singapore. Association for Computational Linguistics.

Antigoni Founta, Constantinos Djouvas, Despoina Chatzakou, Ilias Leontiadis, Jeremy Blackburn, Gianluca Stringhini, Athena Vakali, Michael Sirivianos, and Nicolas Kourtellis. 2018. Large scale crowdsourcing and characterization of twitter abusive behavior. In *Proceedings of the international AAAI conference on web and social media*, volume 12.

Matt Gardner, Yoav Artzi, Victoria Basmov, Jonathan Berant, Ben Bogin, Sihao Chen, Pradeep Dasigi, Dheeru Dua, Yanai Elazar, Ananth Gottumukkala, Nitish Gupta, Hannaneh Hajishirzi, Gabriel Ilharco, Daniel Khashabi, Kevin Lin, Jiangming Liu, Nelson F. Liu, Phoebe Mulcaire, Qiang Ning, Sameer Singh, Noah A. Smith, Sanjay Subramanian, Reut Tsarfaty, Eric Wallace, Ally Zhang, and Ben Zhou. 2020. Evaluating models' local decision boundaries via contrast sets. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 1307–1323, Online. Association for Computational Linguistics.

Jochen Hartmann, Mark Heitmann, Christian Siebert, and Christina Schamp. 2023. More than a feeling: Accuracy and application of sentiment analysis. *International Journal of Research in Marketing*, 40(1):75–87.

Thomas Hartvigsen, Saadia Gabriel, Hamid Palangi, Maarten Sap, Dipankar Ray, and Ece Kamar. 2022. ToxiGen: A large-scale machine-generated dataset for adversarial and implicit hate speech detection. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3309–3326, Dublin, Ireland. Association for Computational Linguistics.

Zexue He, Marco Tulio Ribeiro, and Fereshte Khani. 2023. Targeted data generation: Finding and fixing model weaknesses. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8506–8520, Toronto, Canada. Association for Computational Linguistics.

Yuzhao Heng, Chunyuan Deng, Yitong Li, Yue Yu, Yinghao Li, Rongzhi Zhang, and Chao Zhang.

2024. Proggen: Generating named entity recognition datasets step-by-step with self-reflexive large language models. *arXiv preprint arXiv:2403.11103*.

Ari Holtzman, Jan Buys, Li Du, Maxwell Forbes, and Yejin Choi. 2019. The curious case of neural text degeneration. In *International Conference on Learning Representations*.

Ganesh Jawahar, Muhammad Abdul-Mageed, and Laks Lakshmanan, V.S. 2020. Automatic detection of machine generated text: A critical survey. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 2296–2309, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Karthikeyan K, Shaily Bhatt, Pankaj Singh, Somak Aditya, Sandipan Dandapat, Sunayana Sitaram, and Monojit Choudhury. 2022. Multilingual CheckList: Generation and evaluation. In *Findings of the Association for Computational Linguistics: AACL-IJCNLP 2022*, pages 282–295, Online only. Association for Computational Linguistics.

Nora Kassner and Hinrich Schütze. 2020. Negated and misprimed probes for pretrained language models: Birds can talk, but cannot fly. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7811–7818, Online. Association for Computational Linguistics.

Brendan Kennedy, Mohammad Atari, Aida Mostafazadeh Davani, Leigh Yeh, Ali Omrani, Yehsong Kim, Kris Coombs, Shreya Havaldar, Gwenyth Portillo-Wightman, Elaine Gonzalez, et al. 2018. The gab hate corpus: A collection of 27k posts annotated for hate speech. *PsyArXiv. July*, 18.

Douwe Kiela, Max Bartolo, Yixin Nie, Divyansh Kaushik, Atticus Geiger, Zhengxuan Wu, Bertie Vidgen, Grusha Prasad, Amanpreet Singh, Pratik Ringshia, Zhiyi Ma, Tristan Thrush, Sebastian Riedel, Zeerak Waseem, Pontus Stenetorp, Robin Jia, Mohit Bansal, Christopher Potts, and Adina Williams. 2021. Dynabench: Rethinking benchmarking in NLP. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4110–4124, Online. Association for Computational Linguistics.

Abdullatif Köksal, Renat Aksitov, and Chung-Ching Chang. 2023. Hallucination augmented recitations for language models. *arXiv preprint arXiv:2311.07424*.

Abdullatif Köksal, Timo Schick, Anna Korhonen, and Hinrich Schütze. 2024. Longform: Effective instruction tuning with reverse instructions. *Preprint*, arXiv:2304.08460.

Yukyung Lee, Joonghoon Kim, Jaehee Kim, Hyowon Cho, and Pilsung Kang. 2024. Checkeval: Robust evaluation framework using large language model via checklist. *arXiv preprint arXiv:2403.18771*.

Zhuoyan Li, Hangxiao Zhu, Zhuoran Lu, and Ming Yin. 2023. Synthetic data generation with large language models for text classification: Potential and limitations. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 10443–10461, Singapore. Association for Computational Linguistics.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Rowan Hall Maudslay and Ryan Cotterell. 2021. Do syntactic probes probe syntax? experiments with jabberwocky probing. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 124–131, Online. Association for Computational Linguistics.

Alberto Muñoz-Ortiz, Carlos Gómez-Rodríguez, and David Vilares. 2023. Contrasting linguistic patterns in human and llm-generated text. *arXiv preprint arXiv:2308.09067*.

Humza Naveed, Asad Ullah Khan, Shi Qiu, Muhammad Saqib, Saeed Anwar, Muhammad Usman, Nick Barnes, and Ajmal Mian. 2023. A comprehensive overview of large language models. *arXiv preprint arXiv:2307.06435*.

David Patterson, Joseph Gonzalez, Quoc Le, Chen Liang, Lluis-Miquel Munguia, Daniel Rothchild, David So, Maud Texier, and Jeff Dean. 2021. Carbon emissions and large neural network training. *arXiv preprint arXiv:2104.10350*.

Ethan Perez, Saffron Huang, Francis Song, Trevor Cai, Roman Ring, John Aslanides, Amelia Glaese, Nat McAleese, and Geoffrey Irving. 2022. Red teaming language models with language models. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 3419–3448, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Pranav Rajpurkar, Robin Jia, and Percy Liang. 2018. Know what you don't know: Unanswerable questions for SQuAD. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 784–789, Melbourne, Australia. Association for Computational Linguistics.

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*, pages 6174–6184, Florence, Italy. Association for Computational Linguistics.

Marco Tulio Ribeiro and Scott Lundberg. 2022. Adaptive testing and debugging of NLP models. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3253–3267, Dublin, Ireland. Association for Computational Linguistics.

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.

Rebecca Roelofs. 2019. *Measuring Generalization and overfitting in Machine learning*. University of California, Berkeley.

Anna Rogers, Olga Kovaleva, and Anna Rumshisky. 2020. A primer in BERTology: What we know about how BERT works. *Transactions of the Association for Computational Linguistics*, 8:842–866.

Paul Röttger, Bertie Vidgen, Dong Nguyen, Zeerak Waseem, Helen Margetts, and Janet Pierrehumbert. 2021. HateCheck: Functional tests for hate speech detection models. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 41–58, Online. Association for Computational Linguistics.

Rachel Rudinger, Chandler May, and Benjamin Van Durme. 2017. Social bias in elicited natural language inferences. In *Proceedings of the First ACL Workshop on Ethics in Natural Language Processing*, pages 74–79, Valencia, Spain. Association for Computational Linguistics.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. *arXiv preprint arXiv:1910.01108*.

Maarten Sap, Dallas Card, Saadia Gabriel, Yejin Choi, and Noah A. Smith. 2019. The risk of racial bias in hate speech detection. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1668–1678, Florence, Italy. Association for Computational Linguistics.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D. Manning, Andrew Ng, and Christopher Potts. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1631–1642, Seattle, Washington, USA. Association for Computational Linguistics.

Emma Strubell, Ananya Ganesh, and Andrew McCallum. 2020. Energy and policy considerations for modern deep learning research. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pages 13693–13696.

Antonio Torralba and Alexei A. Efros. 2011. Unbiased look at dataset bias. In *The 24th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2011, Colorado Springs, CO, USA, 20-25 June 2011*, pages 1521–1528. IEEE Computer Society.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. 2023. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

Trieu H. Trinh, Yuhuai Wu, Quoc V. Le, He He, and Thang Luong. 2024. Solving olympiad geometry without human demonstrations. *Nature*, 625(7995):476–482.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc V Le, Ed H. Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. 2023. Self-consistency improves chain of thought reasoning in language models. In *The Eleventh International Conference on Learning Representations*.

Leonie Weissweiler, Abdullatif Köksal, and Hinrich Schütze. 2024. Hybrid human-llm corpus construction and llm evaluation for rare linguistic phenomena. *Preprint*, arXiv:2403.06965.

Chenxi Whitehouse, Monojit Choudhury, and Alham Aji. 2023. LLM-powered data augmentation for enhanced cross-lingual performance. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 671–686, Singapore. Association for Computational Linguistics.

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*, pages 747–763, Florence, Italy. Association for Computational Linguistics.

Guanqun Yang, Mirazul Haque, Qiaochu Song, Wei Yang, and Xueqing Liu. 2022. TestAug: A framework for augmenting capability-based NLP tests. In *Proceedings of the 29th International Conference on Computational Linguistics*, pages 3480–3495, Gyeongju, Republic of Korea. International Committee on Computational Linguistics.

Jiacheng Ye, Jiahui Gao, Qintong Li, Hang Xu, Jiangtao Feng, Zhiyong Wu, Tao Yu, and Lingpeng Kong. 2022. ZeroGen: Efficient zero-shot learning via dataset generation. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 11653–11669, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Yue Yu, Yuchen Zhuang, Jieyu Zhang, Yu Meng, Alexander J. Ratner, Ranjay Krishna, Jiaming Shen, and Chao Zhang. 2023. Large language model as attributed training data generator: A tale of diversity and bias. In *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*.

Xuhui Zhou, Maarten Sap, Swabha Swayamdipta, Yejin Choi, and Noah Smith. 2021. Challenges in automated debiasing for toxic language detection. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 3143–3155, Online. Association for Computational Linguistics.

A Appendix

A.1 Prompts Design

Prompt for sentiment classification task:

Question: Find the sentiment of this text. Answer with positive or negative: that is far too tragic to merit such superficial treatment

Answer: negative

Question: Find the sentiment of this text. Answer with positive or negative: a smile on your face

Answer: positive

Question: Find the sentiment of this text. Answer with positive or negative: saw how bad this movie was

Answer: negative

Question: Find the sentiment of this text. Answer with positive or negative: the greatest musicians

Answer: positive

Question: Find the sentiment of this text. Answer with positive or negative: text

Answer:

Prompt for toxic language detection task:

Determine if the following text includes hate speech or offensive language. Respond with 'Yes' if it contains such language, or 'No' if it is free from hate speech and offensive language.

Text: "japanese or korean people are fucking weird"

Answer: Yes

Text: "the most dangerous job in america is fishing"

Answer: No

Text: "if a girl is raped, it is her fault because she was wearing a short skirt and a low cut blouse and she was walking alone at night"

Answer: Yes

Text: "i'll get to that at the women 's meeting tomorrow"

Answer: No

A.2 Accuracy of Adjectives

We measure the frequency of specific adjectives in sentences where the RoBERTa-Large-based model failed across all test types and find that negative adjectives overall have lower accuracy compared to positive adjectives. The results is shown in Table 4.

	ADJ	Accuracy(%)
POS ADJ	app-ealing	70.2309
	inv-iting	65.2861
	f-avorable	59.8896
	ide-al	57.1285
	joy-ful	56.0241
	en-chant-ing	53.4137
	ch-arming	49.0462
	ex-h-ilar-ating	48.8454
	super	48.5693
	imp-ressive	47.7661
	super-ior	46.2098
	pleasant	46.1345
	ad-mir-able	44.5532
	perfect	42.5201
	g-orge-ous	41.0392
	re-fres-hing	40.0602
	great	39.1315
	ple-asing	39.0311
	am-azing	36.8725
	br-ill-iant	36.8223
	aw-esome	36.1446
	w-onder-ful	35.9689
	s-atisf-ying	35.8685
	inc-redible	35.3916
	del-ight-ful	34.9398
	extra-ordinary	34.739
	fab-ulous	34.5131
	rem-arkable	34.4378
	mar-vel-ous	34.4127
	ex-cellent	33.2329
	f-ant-astic	32.8062
	except-ional	31.3002
	super-b	31.0492
	ter-r-ific	30.9488
	ex-empl-ary	30.246
	out-standing	29.4177
NEG ADJ	bor-ing	29.8008
	du-ll	29.0386
	offensive	21.6622
	medi-ocre	20.9491
	sh-oddy	19.7443
	dis-agree-able	17.5805
	dis-app-oint-ing	17.2363
	un-exc-iting	16.9904
	ter-rible	15.4659
	un-imp-ressive	15.3922
	aw-ful	15.1217
	d-read-ful	14.3349
	un-interesting	14.1628
	app-alling	13.3514
	un-acceptable	13.2776
	hor-rend-ous	12.4908
	pat-hetic	11.2368

Table 4: Accuracy of positive adjectives in sentiment analysis task for RoBERTa-large-based model.

A.3 Ethnic Slurs

We conducted a more comprehensive test of Wikipedia List of Ethnic Slurs, where we use 60 non-toxic sentences as templates to generate sentences to evaluate the performance of the models. The results in Table 5 show that both models are not sensitive enough to many ethnic slurs. Of course, some of the words in the list may have other non-toxic meanings, for example, "`Apple`" can also mean food, and "`Ann`" may really be a person's name.

The accuracy `2.00%` appears frequently, because the single pattern "`{ETHNIC SLURS}'s dog is barking.`" is often incorrectly determined as toxic.

Ethnic Slurs	Accuracy(%)	
	Distilbert	RoBERTa
Abbie	0.00	0.00
ABC	0.00	2.00
ABCD	0.00	2.00
Abid/Abeed	0.00	0.00
Abo/Abbo	0.00	0.00
Afro engineering	56.00	50.00
Ah Chah	0.00	0.00
Ali Baba	0.00	0.00
Alligator bait	2.00	4.00
Alpine Serb	2.00	2.00
AmaLawu	2.00	0.00
Ang mo	0.00	0.00
Ann	0.00	0.00
Annamite, mites	4.00	0.00
Ape	6.00	2.00
Apple	0.00	0.00
Arabush	2.00	0.00
Argie	2.00	0.00
Armo	0.00	0.00
Asing, Aseng	0.00	0.00
Nazi	28.00	92.00
Aunt Jemima	0.00	0.00
Bachicha	0.00	0.00
Baiano	0.00	0.00
Bamboula	0.00	0.00
Banaan	0.00	0.00
Balija	0.00	0.00
Banana	0.00	0.00
Banderite	0.00	2.00
Barbarian	14.00	2.00
Beaner	0.00	0.00
Bimbo	2.00	74.00
Black buck	8.00	0.00
Bluegum	0.00	4.00
Boche	0.00	2.00
Boeotian	0.00	0.00
Boerehater	0.00	0.00
Bog	6.00	0.00
Bogate	2.00	0.00
Bohunk	0.00	0.00
Bong	0.00	0.00
Boong	2.00	0.00
Boonga	0.00	0.00
Bootlip	2.00	0.00
Bougnoule	0.00	0.00
Bounty bar	2.00	0.00
Bozgor	0.00	0.00
Brownie	0.00	0.00
Buckwheat	2.00	0.00
Buddhahead	2.00	0.00
Buckra	0.00	0.00
Bulbash	0.00	0.00
Bule	0.00	0.00
Bumbay	6.00	0.00
Burrhead	2.00	0.00
Bushy	2.00	0.00
Cabbage Eater	4.00	0.00
Canaca	0.00	0.00
Camel jockey	2.00	0.00
Carcamano	0.00	0.00
Chankoro	0.00	0.00
Charlie	0.00	0.00
China Swede	2.00	2.00
Chee-chee	0.00	0.00
Cheese-eating surrender monkeys	58.00	4.00
Chefur (čefur)	0.00	0.00
Tsekwa / Chekwa	0.00	0.00
Chernozhopy	0.00	0.00
Chilote	0.00	0.00
Chinaman	6.00	2.00
Ching chong	0.00	0.00
Chink	50.00	0.00
Chinky	58.00	0.00
Chonky	24.00	0.00
Christ-killer	2.00	30.00
Choc-ice	0.00	0.00
Cholo	0.00	0.00
Chon	0.00	0.00
Chow	6.00	0.00
Chuchmek	0.00	0.00
Chug	0.00	0.00
Chukhna	0.00	0.00
Churka	0.00	0.00
Ciapaty, ciapak	0.00	0.00
Cigányforma	0.00	0.00
Cigány népek	0.00	0.00
Cioară	0.00	0.00
Cina	0.00	0.00
Coconut	0.00	0.00
Pacific Islander	4.00	2.00
Coño	0.00	0.00

continued on next page

continued from previous page

Ethnic Slurs	Accuracy(%)	
	Distilbert	RoBERTa
Coolie	0.00	0.00
Coon	0.00	0.00
Coonass	0.00	0.00
Coreano	0.00	0.00
Cotton picker	2.00	2.00
Cracker	4.00	0.00
Crow	2.00	0.00
Crucco	0.00	0.00
Culchie	0.00	0.00
Curepí	0.00	0.00
Curry-muncher	2.00	2.00
Cushi	0.00	0.00
Czarnuch	0.00	0.00
Dago	0.00	0.00
Dal Khor	0.00	0.00
Dalle, Batak Dalle	0.00	0.00
darky	0.00	0.00
Dhoti	0.00	0.00
Dink	0.00	2.00
Dogan, dogun	2.00	0.00
Dothead, Dot	0.00	0.00
Dune coon	4.00	0.00
Eight ball	0.00	0.00
Engelsman	0.00	0.00
Eyetie	0.00	0.00
Fankui	0.00	0.00
Farang	0.00	0.00
Fenian	8.00	0.00
Festival children	0.00	2.00
verlan	0.00	0.00
Fjellabe	0.00	0.00
Flip	0.00	0.00
Franchute	0.00	0.00
Frenk	0.00	2.00
Fritz	0.00	0.00
Frog	0.00	0.00
Fuzzy-Wuzzy	0.00	0.00
Gabacho	0.00	0.00
Gabel	0.00	0.00
Gadjo	0.00	0.00
Gaijin	0.00	4.00
Galla	0.00	0.00
Gam, Gammat	0.00	0.00
Gans	0.00	0.00
Garoi	0.00	0.00
Geomdung-i	0.00	0.00
Gexhë	0.00	0.00
Gin	0.00	0.00
Gin jockey	2.00	2.00
Godon	0.00	0.00
Golliwog	0.00	0.00
Gook	94.00	0.00
Goombah	4.00	0.00
Gora	0.00	0.00
Goy	0.00	0.00
Grago	6.00	0.00
Greaser	12.00	10.00
Greenhorn	0.00	0.00
Gringo	0.00	8.00
Groid	0.00	2.00
Gub, Gubba	2.00	0.00
Guizi	0.00	0.00
Guido	0.00	0.00
Guinea	2.00	0.00
Gummihals	2.00	0.00
Gusano	0.00	0.00
Gweilo	0.00	0.00
Gwer	0.00	0.00
Gyp/Gip	2.00	0.00
Gyopo, Kyopo	0.00	0.00
Gypsy	0.00	0.00
Hairyback	52.00	0.00
Hajji	0.00	0.00
Half-breed	16.00	4.00
Half-caste	20.00	0.00
Haole	2.00	2.00
Heeb, Hebe	0.00	0.00
Heigui	0.00	0.00
Heukhyeong	0.00	0.00
Hevosmies	0.00	0.00
Hike	0.00	0.00
Hillbilly	0.00	2.00
Honky	0.00	0.00
Hori	0.00	0.00
Hottentot, Hotnot	2.00	0.00
Houtkop	0.00	0.00
Huan-a, Huana	0.00	0.00
Huinca	0.00	0.00
Hujaa	0.00	0.00
Hun	0.00	0.00
Hunky	16.00	0.00
Hymie	0.00	0.00
Ikey	4.00	0.00
Ikey-mo	0.00	0.00

continued on next page

continued from previous page

Ethnic Slurs	**Accuracy(%)**	
	Distilbert	**RoBERTa**
Indon	0.00	0.00
Indognesial	0.00	2.00
Intsik	0.00	0.00
Inyenzi	0.00	0.00
Injun	0.00	0.00
Itaker	0.00	0.00
Jackeen	0.00	0.00
Jakun	0.00	0.00
Jamet	0.00	0.00
Japa	0.00	0.00
Jap	12.00	0.00
Japie	0.00	0.00
Jareer	0.00	0.00
Jerry	0.00	0.00
Jewboy	14.00	4.00
Jidan	0.00	0.00
Jigaboo	2.00	2.00
Jim Crow	6.00	2.00
Jjangkkae	0.00	0.00
Jjokbari	0.00	0.00
Jock	0.00	0.00
Jungle bunny	0.00	0.00
Jutku	2.00	0.00
Kaew	0.00	0.00
Kaffir	0.00	0.00
Kaffir boetie	0.00	0.00
Kalar	0.00	0.00
Kalia	0.00	0.00
Katwa	0.00	0.00
Kanaka	0.00	0.00
Kanake	0.00	0.00
Kano	0.00	0.00
Kaouiche	0.00	0.00
Käskopp	0.00	0.00
Katsap	0.00	0.00
Kebab	0.00	0.00
Keko	0.00	0.00
Keling	0.00	0.00
Kemosabe	0.00	0.00
Kettō	0.00	0.00
Russian	0.00	4.00
Kharkhuwa	0.00	0.00
Khokhol	0.00	0.00
Ikula	0.00	0.00
Kike	0.00	0.00
Kimchi	0.00	0.00
Kıro	0.00	0.00
Knacker	2.00	2.00
Kojaengi	0.00	0.00
Kolorad	0.00	0.00
Krankie	0.00	2.00
Krakkemut	0.00	0.00
Kraut	0.00	0.00
Kuronbō	0.00	0.00
Kkamdungi	0.00	0.00
Labus	0.00	0.00
Laowai	0.00	0.00
Land thief	34.00	4.00
Lapp	0.00	0.00
Lebo, Leb	2.00	0.00
Leupe lonko	2.00	0.00
Limey	0.00	0.00
Locust	6.00	0.00
Londo	0.00	0.00
Lubra	0.00	0.00
Lundy	0.00	0.00
Lugan	0.00	0.00
Mabuno/Mahbuno	0.00	0.00
Macaca	0.00	0.00
Macaronar	0.00	0.00
Majus	0.00	0.00
Malakh-khor	0.00	0.00
Malau	0.00	0.00
Malaun	0.00	0.00
Malingsia	0.00	0.00
Malon	0.00	0.00
Mangal	0.00	0.00
Manne	0.00	0.00
Marokaki	2.00	0.00
Maruta	0.00	0.00
Mau-Mau	0.00	0.00
Mayate/Mayatero	0.00	0.00
Mayonnaise Monkey	2.00	0.00
Mick	0.00	0.00
Mocro	2.00	0.00
Mof	2.00	0.00
Momo	0.00	0.00
Monkey	0.00	0.00
Moskal	0.00	0.00
Moon Cricket	0.00	0.00
Mountain Turk	2.00	0.00
Mulignan	0.00	0.00
Munt	10.00	0.00
Mustalainen	0.00	0.00

continued on next page

continued from previous page

Ethnic Slurs	**Accuracy(%)**	
	Distilbert	**RoBERTa**
Maxhup	0.00	0.00
Mzungu	0.00	2.00
Nawar	0.00	0.00
Neftenya	0.00	0.00
Němčour	0.00	0.00
Nere	0.00	0.00
Niakoué	0.00	0.00
Niglet	0.00	26.00
Nig-nog	0.00	24.00
`N-word`	98.00	92.00
`N-word`itis	88.00	70.00
Nip	20.00	2.00
Nitchie	0.00	0.00
Pribumi	0.00	0.00
Northern Monkey	14.00	0.00
Nusayri	0.00	2.00
Ofay	0.00	0.00
Oláh	0.00	0.00
Orc	0.00	2.00
Oreo	0.00	0.00
Oven Dodger	10.00	0.00
Overner	0.00	0.00
Paddy	0.00	0.00
Pajeet	0.00	0.00
Paki	0.00	6.00
Palagi	0.00	0.00
Paleface	4.00	10.00
Pancake Face	0.00	0.00
Papoose	0.00	0.00
Paraíba	0.00	0.00
Parsubang	0.00	0.00
Pastel de flango	0.00	0.00
Peckerwood	0.00	0.00
Peenoise	2.00	0.00
Perker	6.00	0.00
Pepper or Pepsi	0.00	0.00
Pickaninny	2.00	0.00
Piefke	0.00	0.00
Pikey	0.00	0.00
Pindos	0.00	0.00
Pink pig	54.00	32.00
Plastic Paddy	4.00	2.00
Plouc	0.00	0.00
Pocho	0.00	0.00
Pocahontas	0.00	0.00
Polack	0.00	0.00
Polaco	0.00	0.00
Polaca	0.00	0.00
Polentone	0.00	0.00
Pommy	4.00	0.00
Porridge wog	4.00	0.00
Portagee	0.00	0.00
Potet	0.00	0.00
Prairie `N-word`	86.00	94.00
Prod	2.00	0.00
Pshek	0.00	0.00
Quashie	2.00	0.00
Raghead	30.00	0.00
Ramasamy	0.00	0.00
Rastus	14.00	0.00
Razakars	0.00	0.00
Redlegs	10.00	0.00
Redskin	2.00	2.00
Risorse boldriniane	0.00	0.00
Rockspider	0.00	0.00
Rootless cosmopolitan	2.00	2.00
Rosuke	0.00	0.00
Rooinek	0.00	0.00
Roto	0.00	0.00
Roundeye	2.00	2.00
Russki	0.00	0.00
Safavid	0.00	0.00
Sambo	0.00	2.00
Sand `N-word`	100.00	98.00
Sangokujin	0.00	0.00
Sarong Party Girl	0.00	0.00
Sassenach	0.00	0.00
Savage	22.00	0.00
Sawney	0.00	0.00
Scandihoovian	0.00	2.00
Seppo, Septic	8.00	0.00
Schluchtenscheißer	0.00	0.00
Schvartse	0.00	0.00
Schwartze Khayeh	0.00	0.00
Sibun River	2.00	0.00
Sheeny	0.00	0.00
Sheepshagger	64.00	0.00
Shelta	0.00	0.00
Shegetz	0.00	0.00
Shina	0.00	0.00
Zhina	0.00	0.00
Shine	0.00	0.00
Shiptar	0.00	0.00
Shka i Velikës	0.00	0.00

continued on next page

continued from previous page

Ethnic Slurs	Accuracy(%)	
	Distilbert	RoBERTa
Shkije	0.00	0.00
Shkinulkë	0.00	0.00
Shkutzim	2.00	0.00
ShkutorCroatian	0.00	0.00
Shoneen	0.00	0.00
Shylock	0.00	0.00
Sideways vagina	80.00	100.00
Skinny	4.00	2.00
Skopianoi	0.00	0.00
Skip, Skippy	0.00	0.00
Skævøjet	0.00	0.00
Slant	10.00	2.00
Slobo	8.00	0.00
Slope	4.00	0.00
Snowflake	0.00	0.00
Smoked Irish/Smoked Irishman	16.00	2.00
Somdeang	0.00	0.00
Somkhao	0.00	0.00
Soosmar-khor	0.00	0.00
Sooty	18.00	2.00
Southern Faerie	2.00	2.00
Soutpiel	0.00	0.00
Spade	0.00	0.00
Spearchucker	4.00	0.00
Spic	0.00	0.00
Spook	6.00	0.00
Squarehead	2.00	0.00
Squaw	2.00	0.00
Swamp Guinea	34.00	0.00
skopčák	0.00	0.00
Szwab	0.00	0.00
Taffy	0.00	0.00
Taig	0.00	0.00
Tai Ke	0.00	0.00
Tanka	0.00	0.00
Tar-Baby	34.00	0.00
Tàu	0.00	0.00
Teabag	0.00	2.00
Teapot	0.00	0.00
Terrone	0.00	0.00
Teuchter	0.00	0.00
Thicklips	2.00	0.00
Tibla	0.00	0.00
Tiko	0.00	0.00
Timber N-word	92.00	98.00
Timur	0.00	0.00
Ting tong	0.00	2.00
Tinker	0.00	0.00
Toad	0.00	0.00
Toku-A	0.00	0.00
Tonto	0.00	0.00
Touch of the tar brush	0.00	0.00
Towel head	2.00	2.00
Turco-Albanian	6.00	2.00
Turco	2.00	2.00
Turčin, Poturčin	0.00	0.00
Turk	0.00	0.00
Turko	0.00	2.00
Twink	74.00	2.00
Ukro-Nazi	48.00	88.00
Ukrop	0.00	0.00
Uncle Tom	0.00	0.00
Unta	0.00	0.00
UPAina	0.00	0.00
Uppity	0.00	36.00
Uzkoglazyj	0.00	0.00
Vanja	0.00	0.00
Veneco	0.00	0.00
Vrindavan	0.00	0.00
Vuzvuz	0.00	0.00
Wagon burner	2.00	0.00
Wasi'chu	0.00	0.00
West Brit	0.00	2.00
Wetback	18.00	0.00
White ears	20.00	0.00
White interloper	20.00	12.00
Wigger	4.00	30.00
White N-word/N-word wop	100.00	100.00
White trash	44.00	58.00
Whitey	0.00	4.00
Wog	10.00	0.00
Wop	6.00	0.00
Xiǎo Rìběn	0.00	0.00
Xing Ling	0.00	0.00
Yam yam	0.00	0.00
Yanacona	0.00	0.00
Yank	0.00	12.00
Yankee	2.00	2.00
Yaposhka	0.00	2.00
Yellow	2.00	0.00
Yellow bone	6.00	2.00
Yid	0.00	0.00
Yuon	0.00	0.00
Zip, Zipperhead	2.00	0.00

continued on next page

continued from previous page

Ethnic Slurs	Accuracy(%)	
	Distilbert	RoBERTa
Zuca, Brazuca	0.00	0.00
Zhyd	0.00	0.00

Table 5: Accuracies of SiEBERT and DistilBERT in testing the list of ethnic slurs.

A.4 Amount of Sentences

Different templates contain different placeholders will result in different amounts of sentences generated from the templates, for reference, we list the number of sentences corresponding to each template in the two tasks in Table 6 and Table 7. It can be seen that the correlation between the number of sentences and the accuracy is not obvious.

Test Type	Template	Amount
Negation	This { $\text{book}_{\text{NOUN}}$ } is not { $\text{awful}_{\text{NEG ADJ}}$ $\text{nice}_{\text{POS ADJ}}$ }.	1411 2988
	I don't think this { $\text{book}_{\text{NOUN}}$ } is { $\text{awful}_{\text{NEG ADJ}}$ $\text{nice}_{\text{POS ADJ}}$ }.	1411 2988
	It isn't true that this { $\text{book}_{\text{NOUN}}$ } isn't { $\text{nice}_{\text{POS ADJ}}$ $\text{awful}_{\text{POS ADJ}}$ }.	2988 1411
	I can't find anything { $\text{awful}_{\text{NEG ADJ}}$ $\text{nice}_{\text{POS ADJ}}$ } to say about this { $\text{book}_{\text{NOUN}}$ }.	1411 2988
	I am unable to find anything { $\text{awful}_{\text{NEG ADJ}}$ $\text{nice}_{\text{POS ADJ}}$ } to say about this { $\text{book}_{\text{NOUN}}$ }.	1411 2988
	I don't find anything { $\text{awful}_{\text{NEG ADJ}}$ $\text{nice}_{\text{POS ADJ}}$ } to say about this { $\text{book}_{\text{NOUN}}$ }.	1411 2988
Past Tense	{ $\text{I was wrong.}_{\text{REVISION}}$ } I thought this { $\text{book}_{\text{NOUN}}$ } was { $\text{awful}_{\text{NEG ADJ}}$ $\text{nice}_{\text{POS ADJ}}$ }.	12699 26892
	I thought this { $\text{book}_{\text{NOUN}}$ } was { $\text{awful}_{\text{NEG ADJ}}$ $\text{nice}_{\text{POS ADJ}}$ }. { $\text{I was wrong.}_{\text{REVISION}}$ }	12699 26892
Comparative	I'm sure I'll see plenty in the future, but I'm sure none will be as { $\text{nice}_{\text{POS ADJ}}$ $\text{awful}_{\text{NEG ADJ}}$ } as this { $\text{book}_{\text{NOUN}}$ }.	2988 1411
	There can't be any { $\text{worse}_{\text{NEG COMPARATIVE ADJ}}$ $\text{better}_{\text{POS COMPARATIVE ADJ}}$ } { $\text{book}_{\text{NOUN}}$ } than this one.	83 83
Specific Phrase	I was blown away by this { $\text{book}_{\text{NOUN}}$ }.	83
	This { $\text{book}_{\text{NOUN}}$ } is a perfect little atrocity...	83

Table 6: The amount of sentences for each template in sentiment analysis task, where each template represents different linguistic structures and test scenarios.

Test Type	Template	Amount
Stereotypes and Directives	{ Asians$_{\text{NATIONALITY}}$ You$_{\text{PERSONAL PRONOUN}}$ } { deserve to$_{\text{TERM OF “DESERVE TO”}}$ } die.	102 24
	{ Asians$_{\text{NATIONALITY}}$ } should { be restricted from traveling freely$_{\text{DENY HUMAN RIGHTS}}$ }.	1360
	{ Asians$_{\text{NATIONALITY}}$ } should not { be restricted from traveling freely$_{\text{DENY HUMAN RIGHTS}}$ }.	1360
Typos	{ Asians$_{\text{NATIONALITY}}$ } { deseve to$_{\text{“DESERVE” TERM TYPO}}$ } die.	5100
Reversal	{ Asians$_{\text{NATIONALITY}}$ } deserve to die.{ That’s wrong!$_{\text{DENY PHRASE}}$ }	408
Nonsense Characters	{ Asians$_{\text{NATIONALITY}}$ } deserve to die.{ ——>————->——/n$_{\text{STRANGE SUFFIX}}$ } { ——>————->——/n$_{\text{STRANGE PREFIX}}$ } { Asians$_{\text{NATIONALITY}}$ } deserve to die.	3400 3400
Ethnic Slurs	{ It seems this `N-word` forgot his lunch today.$_{\text{NON-TOXIC SENTENCE}}$ } { It seems this Ching chong forgot his lunch today.$_{\text{NON-TOXIC SENTENCE}}$ }	20 20
Sentiment Analysis Pattern	This { book$_{\text{NOUN}}$ } is not { nice$_{\text{POS ADJ}}$ }.	2988
	This { book$_{\text{NOUN}}$ } is { awful$_{\text{NEG ADJ}}$ }.	1411

Table 7: The amount of sentences for each template in sentiment toxic language detection task, where each template represents different linguistic structures and test scenarios.