Benchmarking Sentiment Classifiers Against Adversarial and Stress Attacks

Author: Nikhil Dodda

Abstract

The resilience of lightweight sentiment classification model DistilBERT against real-world adversarial and stress conditions is examined in this work. Typographical mistakes (typo attacks), long junk inputs (flood attacks), and fast-fire input bursts (simulated DDoS) three different kinds of input attacks were assessed. Under clean and adversarial conditions, the evaluation gauges response times as well as classification accuracy. Results highlight the need of adversarial-aware training and system-level protections since DistilBERT is robust to small input noise but deteriorates greatly under stress.

1. Introduction

Sentiment analysis, chatbots, and customer feedback pipelines all make extensive use of contemporary NLP models like BERT and its derivatives. However, operational resilience and input-level robustness are frequently overlooked in favor of accuracy in most of the research.

The goal of this project is to evaluate DistilBERT-base-uncased-finetuned-sst-2-english, a frequently used transformer, under actual attack conditions. Combining sequence flooding, rapid input stress, and text corruption (typos), we model typical threats encountered by deployed systems and examine both model behavior and system performance.

2. Datasets

Two datasets were chosen to represent long-form and short-form text use cases:

- **IMDB Movie Reviews**
  Long, detailed natural reviews. Useful for testing typo robustness with context.
- **Amazon Mobile Apps Reviews**
  Short, high-frequency user reviews. Useful for simulating production-scale stress and low-context inputs.

This dual-dataset approach helps examine how input length and context affect the model's vulnerability.

3. Methodology

3.1 Model

We used **DistilBERT-base-uncased**, fine-tuned on the SST-2 binary sentiment classification task. It is 40% smaller and 60% faster than BERT, making it suitable for production environments.

3.2 Attack Types

Three attack types were implemented:

- **Typo Attack**: Randomly replace or insert characters in each review.
- **Flood Attack**: Append ~1000 characters of meaningless junk text to simulate input flooding.
- **Simulated DDoS**: Rapidly submit 100+ short junk inputs to simulate inference overload.

All attacks were automated using Python and HuggingFace Transformers. Latency was measured using `time.perf_counter()`.

4. Experimental Setup

- **Device**: MacBook Pro (Apple M4 Pro, 24GB RAM)
- **Frameworks**: HuggingFace Transformers, PyTorch, PySpark
- **Visualization**: Matplotlib for response and accuracy graphs

Metrics were captured for:

- Clean and adversarial accuracy
- Average inference time across scenarios
- Behavior under scaling (e.g., truncation or silent failure)

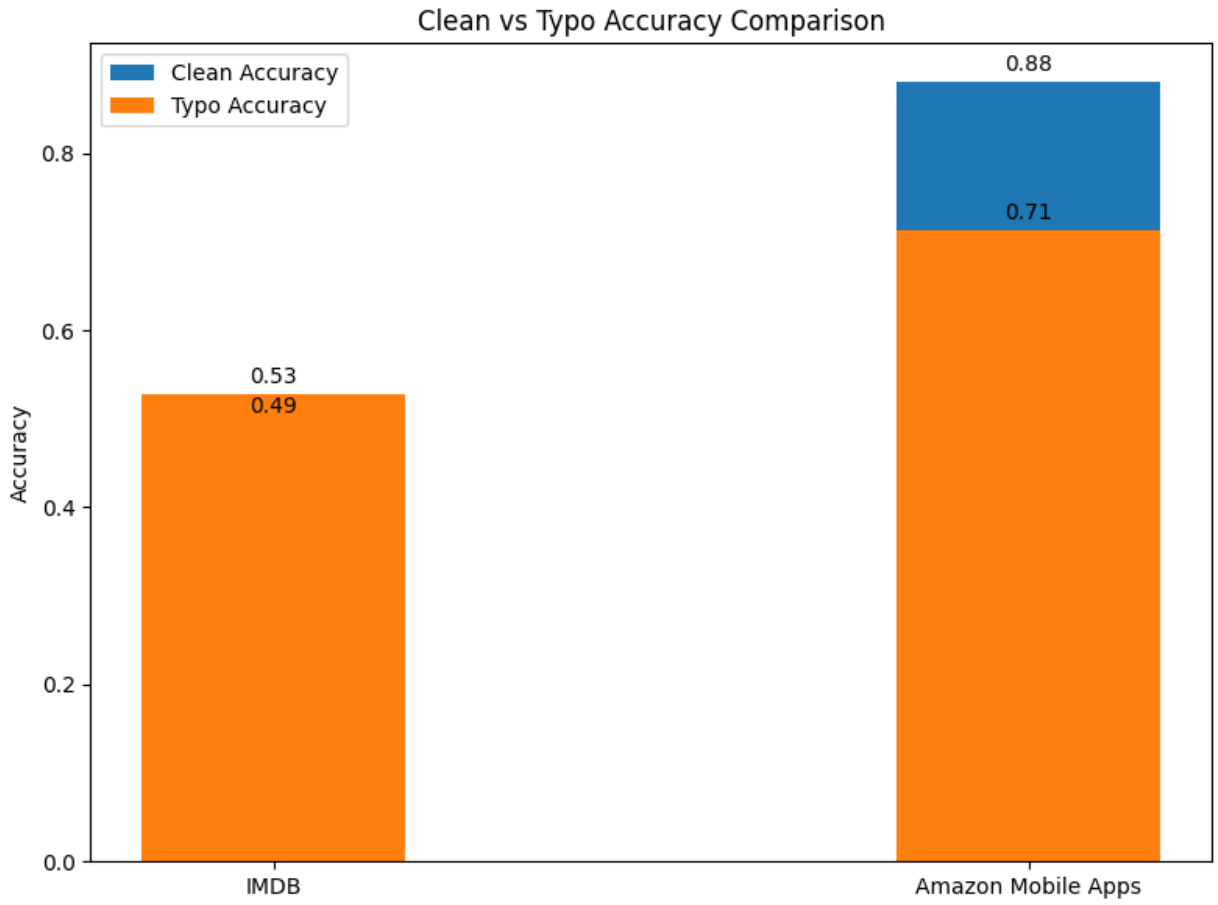
5. Results and Analysis

5.1 Accuracy Table

Dataset	Clean Accuracy	Typo Accuracy
IMDB	0.494	0.528
Amazon Mobile App	0.880	0.713

- On IMDB, typos had minimal impact due to contextual redundancy.
- On Amazon, accuracy dropped ~17% short inputs amplify the effect of corruption.

5.2 Accuracy Graph



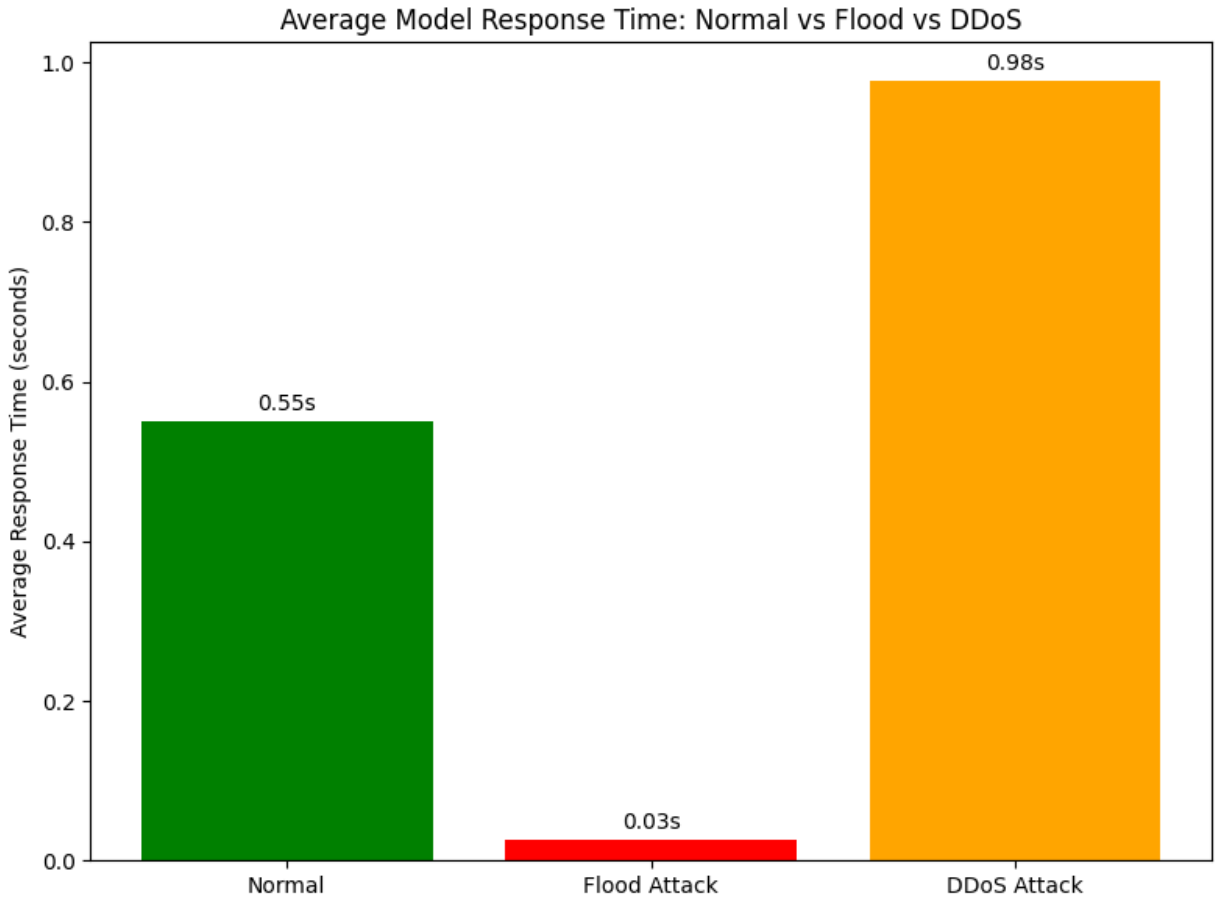
The graph shows that the model is more sensitive to noise when context is limited, emphasizing the importance of robustness in low-context environments.

5.3 Response Timetable

Scenario	Avg. Response Time (s)
Clean Input	0.55
Flood Attack	0.30 (truncated input)
DDoS Attack	0.98

- Flooded inputs caused truncation, reducing actual processing time but also skipping valuable tokens.
- DDoS scenarios led to **nearly 2x latency**, highlighting the model's vulnerability to overload.

5.4 Response Time Graph



6. Discussion

Key takeaways:

- **Typo attacks** greatly reduce performance on short inputs but have little effect on long texts.
- **Flood attacks** use token limits to induce truncation. Silent accuracy loss with no system warning follows from this.
- **DDoS stress** exposes latency sensitivity. While model outputs remain valid, delay raises real-time app risk of timeout.
- DistilBERT shows a blind spot in transformer-based NLP systems by not having any inherent mechanisms to detect or react to overload.

7. Conclusion and Future Work

DistilBERT demonstrates reasonable resilience to casual input noise but struggles under real-world stress patterns.
For future work, I suggest:

- **Adversarial data augmentation** to improve typo tolerance.
- **Input validation** or **rate limiting** at the system layer to mitigate stress conditions.
- **Entropy or confidence-based detection** to identify uncertain or adversarial inputs.

Overall, this benchmark exposes important weaknesses in resilience at the infrastructure and model levels that need to be fixed before implementing NLP systems in high-stakes, user-facing settings.

8. References

1. J. Devlin et al., “BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding,” NAACL 2019.
2. V. Sanh et al., “DistilBERT: A distilled version of BERT,” arXiv:1910.01108.
3. Amazon Reviews Dataset: https://nijianmo.github.io/amazon/index.html
4. IMDB Sentiment Dataset: https://ai.stanford.edu/~amaas/data/sentiment/
5. HuggingFace Transformers: https://huggingface.co/transformers/