

Imperial College of Science, Technology and Medicine
Department of Computing

Robustness in Natural Language Processing

Zhenhao Li

Supervisors: Lucia Specia, Marek Rei

Submitted in part fulfilment of the requirements for the degree of
Doctor of Philosophy in Computing of Imperial College London and
the Diploma of Imperial College, Jul 2025

Statement of Originality

I, Zhenhao Li, hereby declare that this thesis “Robustness in Natural Language Processing”, and the research presented in it are my original work, and the collaborators and work of others have been properly acknowledged and referenced.

Copyright Declaration

The copyright of this thesis rests with the author. Unless otherwise indicated, its contents are licensed under a Creative Commons Attribution-Non Commercial-No Derivatives 4.0 International Licence (CC BY-NC-ND).

Under this licence, you may copy and redistribute the material in any medium or format on the condition that; you credit the author, do not use it for commercial purposes and do not distribute modified versions of the work.

When reusing or sharing this work, ensure you make the licence terms clear to others by naming the licence and linking to the licence text.

Please seek permission from the copyright holder for uses of this work that are not included in this licence or permitted under UK Copyright Law.

Abstract

The recent success in Natural Language Processing has been driven by the evolution of pre-trained language models, which achieve state-of-the-art performance across a wide range of tasks, such as machine translation, sentiment analysis, question answering, etc. Despite these successes, ensuring robustness of NLP systems remains a significant challenge, especially when faced with non-standard inputs such as noisy texts and adversarial texts. In this thesis, we focus on three common robustness issues in NLP systems: noisy texts, textual adversarial attacks, and unintended data bias. We address the robustness issue from textual noise by integrating extra visual and conversational contexts.

We first incorporate visual contexts and error correction training in multimodal translation. Our findings reveal that the integration of multimodality and error correction effectively mitigates information loss from noise and improves translation robustness. In addition, we extend the exploration of noise robustness to multimodal conversational modeling by proposing a new task: multimodal conversation derailment detection and curating a corresponding new dataset. We propose a multimodal hierarchical model that combines textual, visual, and conversational contexts to predict conversation derailment, and experimentally demonstrate the efficacy of conversational contexts in noisy settings. To counter adversarial attacks, we propose a flexible method, DiffuseDef, to defend against adversarial attacks. Our method trains a diffusion layer on top of a PLM and performs denoising iteratively on the encoder hidden representations. Benefiting from removing latent noise in adversarial texts, our method achieves state-of-the-art performance against common black-box and white-box adversarial attacks. Finally, we tackle the robustness concern that stems from model training, specifically handling unintended bias. We propose an adversarial multitask model that extracts task-common representations and disentangles task-specific representations that can lead to bias. Our experiments on toxicity detection show that the proposed method can successfully mitigate unintended biases arising from three relevant tasks.

Acknowledgements

Going through the PhD is a challenging but rewarding experience. Looking back on the last few years, I was lucky to have met so many great people, learned a lot of brilliant research work, and contributed my tiny bit to the NLP community, all of the experience helped me grow and shaped me into who I am. Here I would like to acknowledge the people who helped me throughout the years in my research and in my life.

Firstly, I would like to thank my supervisor, Prof. Lucia Specia. I first met Lucia in 2018 when I was an MSc student at Imperial College London, and Lucia was my supervisor. Lucia is a great supervisor, and I feel very fortunate to have worked with her and learned from her. Her passion for research, expertise in ML/NLP, and invaluable guidance contributed to my growth in research and illuminated the path of my career in NLP.

I would also like to thank my second supervisor, Dr. Marek Rei, for his insightful advice and constructive feedback throughout my PhD. Marek witnessed each stage of this thesis from the early stage assessment to the dissertation, and gave me countless advice that helped improve my research.

Furthermore, I would like to thank my co-workers Dr. Marina Fomicheva, Dr. Ozan Caglayan, Dr. Khetam Al Sharou, and friends in the Language and Multimodal AI Lab (LAMA) group, Zixu Wang, Nihir Vedda, Alexander Gaskell, Antigoni-Maria Founta, Joe Stacey, Atijit Anuchitanukul, Huichi Zhou, etc. These fantastic colleagues are excellent researchers and have been supportive and productive in our collaborations. Without their effort in our collaboration, I would not be able to complete this thesis.

Finally, I here express my heartfelt gratitude to my family, who has been my steadfast and reassuring support throughout my life. I want to thank my parents, Xiaodong Li and Xiaojie Qu, and their love and encouragement are my greatest motivation, which helped me go through the hardest time in my PhD. I would also like to thank my wife, Wenqing Peng, for her love, support, and company. Words can never express all my love and gratitude for you. And to our dog, Xiaoshui, thank you for filling our days with laughter, energy, and countless happy moments that made the PhD journey so much brighter.

Dedication

To my beloved parents, Xiaodong and Xiaojie, who have always guided, inspired, and supported me throughout my life.

‘Talk is cheap. Show me the code.’

Linus Torvalds

Contents

Abstract	iii
Acknowledgements	v
1 Introduction	1
1.1 Motivation	1
1.2 Objectives	4
1.3 Contributions	6
1.4 Publications	7
1.4.1 Related Publications	7
1.4.2 Other Publications	8
1.4.3 Other Research Contributions	9
2 Background	11
2.1 Deep Learning Models	11
2.1.1 Transformer	12
2.1.2 BERT and Other Pre-trained Language Models	16
2.1.3 Denoising Diffusion Probabilistic Model	18

2.2	Robustness Issues in NLP	21
2.2.1	Handling Textual Noise	21
2.2.2	Defending Adversarial Attack	26
2.2.3	Mitigating Unintended Bias	31
3	Multimodal Error Correction for Robustness against Textual Noise	33
3.1	Introduction	34
3.2	Related Work	36
3.3	Methods	38
3.3.1	Multimodal Cross-attention	38
3.3.2	Noise Injection	39
3.3.3	Error Correction Training	41
3.4	Experiments	42
3.4.1	Datasets	42
3.4.2	Models	43
3.5	Results	44
3.5.1	Models Trained on Clean Data	44
3.5.2	Models Trained on Noisy Data	45
3.6	Analysis	49
3.6.1	Robustness on Unseen Noise	49
3.6.2	Visual Sensitivity	50
3.6.3	Effect of λ	51
3.6.4	Error Correction Quality	52

3.7	Qualitative Examples	53
3.8	Conclusions	56
4	Better Noise Robustness with Multimodal Conversation Modeling	57
4.1	Introduction	58
4.2	Related Work	60
4.2.1	Conversation Derailment Detection	60
4.2.2	Conversation Modeling	61
4.3	Multimodal Conversation Derailment Detection	62
4.3.1	Task Definition	62
4.3.2	Data Collection	63
4.3.3	Annotation Procedure	64
4.3.4	Statistics	65
4.4	Model	68
4.4.1	Pre-trained Text Encoder	69
4.4.2	Visual Encoder	69
4.4.3	Title-image Fusion	69
4.4.4	Context Encoder	70
4.4.5	Late Fusion	70
4.5	Experiments	71
4.5.1	Baselines	71
4.5.2	Pre-processing and Training	72
4.6	Results	73

4.6.1	Benchmark	73
4.6.2	Ablation Analysis	75
4.6.3	Model Performance on Different Derailment Reasons	76
4.6.4	MMConv with Different Pre-trained Text/Visual Encoders	76
4.6.5	Forecasting Future Derailment	77
4.7	Robustness Analysis	79
4.7.1	Controlled Noise Injection	79
4.7.2	Robustness to Adversarial Attack	80
4.8	Conclusions	81
5	Improved Robustness to Adversarial Attacks via Iterative Denoising	83
5.1	Introduction	84
5.2	Related Work	86
5.3	DiffuseDef	87
5.3.1	Training	87
5.3.2	Inference	89
5.4	Experiments	90
5.5	Results and Analysis	93
5.5.1	Adversarial Robustness	93
5.5.2	Robustness against White-box Attacks	95
5.5.3	Ablation Analysis	96
5.5.4	DiffuseDef + Adversarial Augmentation	96
5.5.5	Robustness w.r.t Token Length	97

5.5.6	Effect of Additional Denoising Steps	98
5.5.7	Ensembling Diffused Hidden Representations	99
5.5.8	Efficiency of DiffuseDef	101
5.5.9	Example of Noising and Denoising in DiffuseDef	102
5.6	Conclusions	104
6	Adversarial Multi-task Training for Debaised Toxicity Detection	105
6.1	Introduction	106
6.2	Related Work	108
6.2.1	Toxicity Detection with MTL	108
6.2.2	Disentangling Representations	109
6.3	Method	110
6.4	Experiments	113
6.4.1	Data	113
6.4.2	Training and Evaluation	116
6.5	Debiasing Toxicity with Sentiment	117
6.6	Debiasing Toxicity with Profanity	118
6.7	Debiasing Toxicity with Identities	123
6.8	Conclusions	125
7	Conclusions	126
7.1	Summary of Achievements	126
7.2	Future Directions	128
7.2.1	Efficient and Robust NLP	128

7.2.2	LLM Red Teaming	129
7.2.3	Persistable Robustness in Models	130

Bibliography		130
---------------------	--	------------

List of Tables

3.1	An example of noise injected	40
3.2	Noise evaluation of Transformer models trained on clean data	46
3.3	Noise evaluation of Transformer models trained on synthetically noisy data . . .	46
3.4	Effect of noisy training on clean data (Transformer models)	48
3.5	Noise evaluation of GRU models trained on clean data	48
3.6	Effect of noisy training on clean data (GRU models)	49
3.7	Noise evaluation under unknown noise levels	49
3.8	Error correction performance	52
3.9	Qualitative examples for translation and error correction	53
4.1	Types of noise in MCD dataset	67
4.2	Size of the train/dev/test splits for MCD dataset	68
4.3	Multi-modal conversation derailment detection results	74
4.4	Ablation analysis for MMConv	75
4.5	MMConv with different text and visual encoders	77
4.6	Derailment forecasting performance	78
4.7	Adversarial robustness of conversational models	81

5.1	Dataset statistics for adversarial evaluation	90
5.2	Evaluation parameters for each dataset	91
5.3	Hyperparameters for diffusion training	93
5.4	Black-box adversarial robustness results	94
5.5	White-box adversarial robustness results	96
5.6	Ablation analysis for DiffuseDef	96
5.7	Similarity between clean and adversarial hidden states	100
5.8	Efficiency comparison of DiffuseDef and other methods	101
5.9	MLM outputs after noising and denoising	102
5.10	MLM outputs for clean/adv texts after denoising	103
6.1	Examples of biases from relevant tasks in toxicity detection	108
6.2	Statistics of toxicity/sentiment/profanity/identity datasets	115
6.3	Hyperparameters for the MHS dataset	116
6.4	Hyperparameters for the HF dataset	116
6.5	Hyperparameters for the UB dataset	117
6.6	Toxicity results on MHS test set	118
6.7	Toxicity results on the in-domain HF test set	120
6.8	Toxicity results on the OOD Implicit Hate test set	120
6.9	Bias evaluation results on the OOD Implicit Hate test set	121
6.10	Average attribution values for top profanity words	121
6.11	Most important profanity words that contribute to a model toxic prediction . . .	122
6.12	Toxicity results on the UB test set	123

6.13 Top identity words and their word importance scores 124

List of Figures

1.1	Overall contributions of this thesis	6
2.1	Architecture of transformer model	13
2.2	Scaled dot-product attention	14
2.3	Pre-training of BERT	16
2.4	Diffusion and reverse diffusion process in DDPM	19
2.5	Categorizations of noise in NLP	22
2.6	Illustration of adversarial attack pipeline	28
3.1	MMT model can translate unknown words, but fails with noisy input	35
3.2	Joint training of translation and error correction	41
3.3	MMT model outperforms NMT model on noisy data	45
3.4	Incongruent decoding analysis	50
3.5	Effect of λ on translation and error correction tasks	51
3.6	Attention map for the translation decoder	54
3.7	Attention map for the error correction decoder	55
4.1	An example of multimodal conversation derailment detection	59
4.2	Example of different types of derailments	63

4.3	Annotation interface on Amazon Mechanical Turk	65
4.4	Distribution of conversation length in the MCD dataset	66
4.5	Distribution of derailment reasons in the MCD dataset	67
4.6	MMConv architecture	68
4.7	Binary accuracy on test examples with different derailment reasons	77
4.8	Forecast horizon distribution for MMConv model	78
4.9	Robustness analysis on noisy test set	80
5.1	Training and inference of DiffuseDef	87
5.2	Robustness of DiffuseDef with textual adversarial augmentation	97
5.3	Adversarial robustness on texts with different lengths	98
5.4	Adversarial robustness with different denoising step	99
5.5	Distribution of max token importance score	100
6.1	Illustration of the adversarial multi-task model	111
6.2	Visualization of the task-common/specific features using t-SNE	119
6.3	Bias evaluation results on the UB dataset	124

Chapter 1

Introduction

1.1 Motivation

Natural Language Processing (NLP) has gone through remarkable development driven by a series of advances in deep learning, from attention (Bahdanau et al., 2015) and transformer (Vaswani et al., 2017) architectures to pre-trained language models (Devlin et al., 2019) and foundation models (Bommasani et al., 2021). Pre-trained language models, for example, BERT (Devlin et al., 2019), GPT (Radford and Narasimhan, 2018), T5 (Raffel et al., 2020), etc., have been trained on vast amounts of textual data with self-supervised learning (e.g., masked language modeling, next token prediction, etc.) to tackle natural language problems. These models set new benchmarks for various NLP tasks, including machine translation, question answering, sentiment analysis, hate speech detection, etc., achieving and outperforming human performance in many tasks (Rajpurkar et al., 2018; Wang et al., 2019). Their ability to process large amounts of data and learn complex patterns has enabled real-world applications in fields such as conversational AI, content moderation, etc.

Despite the significant success of pre-trained models, these NLP systems are still faced with critical challenges in robustness, which may hinder their reliability in real-world applications, and can lead to severe consequences. For example, automated moderation systems that do not detect toxic language on social media platforms can cause emotional harm to the user or

spread misinformation (Fortuna and Nunes, 2018a); Unreliable machine translation models can lead to serious misunderstandings in sensitive domains such as healthcare and legal systems (Khayrallah and Koehn, 2018). Therefore, it is a crucial topic to study methods to improve the robustness of real-world NLP systems.

Robustness failures can be considered as manifestations of the model’s vulnerabilities, where systematic weaknesses can emerge from the distribution discrepancies between its training and production/evaluation environments. The vulnerability of the model can be classified into two categories based on the source of the vulnerability: **induced vulnerability** and **learned vulnerability**. Induced vulnerabilities are triggered at inference time with non-standard user input (e.g., noisy texts, adversarial examples) to the model, which can deviate from the standard form of data on which the model has been trained (Wang et al., 2022b; Vassilev et al., 2024). Such non-standard inputs may fall out of the distribution of the training data, thus resulting in incorrect outputs. In contrast, the learned vulnerabilities are obtained during the model training, when the model learns spurious correlations or “shortcuts” to the target labels. These correlations or inductive biases do not reflect the real-world distribution, and therefore when evaluated, the model may reproduce unreliable outputs (McCoy et al., 2019; Ye et al., 2024).

Induced vulnerabilities One of the most prevalent induced vulnerabilities can arise from *textual noise* (Wang et al., 2022b). Unlike carefully curated datasets that have been filtered and cleaned, user-generated texts on the Internet are often noisy and exist with diverse language usage, including misspellings, grammatical errors, leet speak, code-switching, etc. Such noisy texts can significantly degrade model performance as they are trained on clean and curated datasets. In contrast to humans, most NLP models are trained on a large number of texts and thus learn strong abilities to memorize previous training examples and generalize to similar texts. However, the number of language variants can be infinite, and the training data cannot cover all possibilities. An interesting illustration is the “Cmabrigde” effect (Davis, 2003):

*Aoccdrnig to a rscheearch at Cmabrigde Uinervtisy, it deosn’t mtttaer in waht oredr
the ltteers in a wrod are, the olny iprmoetnt tihng is taht the frist and lsat ltteer be*

*at the rghit pclae. The rset can be a toatl mses and you can sitll raed it wouthit
porbelm. Tihs is bcuseae the huamn mnid deos not raed ervey lteter by istlef, but
the wrod as a wlohe.*

Although a human can recognize and understand the above texts without challenge, NLP models such as BERT may fail to capture the meaning of disordered sentences (Belinkov and Bisk, 2018). This highlights the limitation of pre-trained NLP models that rely on token-level embeddings. Minor perturbations in the input text can cause a different token distribution from the pre-trained data, resulting in performance degradation.

Furthermore, in addition to the naturally occurring noise in texts, another critical challenge to robust and applicable NLP systems is the threat posed by **adversarial attacks**. In real-world applications, users can input any text into these models, and some may even deliberately craft perturbations to exploit model vulnerability. Adversarial examples often involve subtle modifications, such as synonym substitutions, paraphrasing, etc., that are imperceptible to humans but can fool the model to generate incorrect outputs (Michel et al., 2019). In extreme cases, such attacks can be iterative, where the user repeatedly creates different variants to test until the system fails. For example, AI assistants like Alexa and Siri can be easily fooled by replacing certain words in a sentence with synonyms¹. These failures can even be amplified by public dissemination of successful attacks on social media, which may undermine trust in the model's reliability and integrity.

Learned vulnerabilities On the other hand, robustness issues may arise from the model itself rather than malicious user inputs, one of the most severe learned vulnerabilities being the **unintended bias** in the model. Although models are trained with cleaned and filtered datasets, they might still learn unintended bias from the lexical correlation in the data (Garg et al., 2023). For example, toxicity detection models can associate their prediction with the presence of a specific identity group in texts (Adams et al., 2019), regardless of whether the

¹<https://www.technologyreview.com/2020/02/07/349027/software-that-swaps-out-words-can-now-fool-the-ai-behind-alexa-and-siri/>

context is neutral or not. Such biases are not only considered technical shortcomings, but can also perpetuate human stereotypes and reinforce inequities.

To address the above model vulnerabilities that hinder robustness, researchers have focused on improving robustness by building noisy training and evaluation datasets (Michel and Neubig, 2018), developing techniques to improve model resilience (Attanasio et al., 2022; Cooper Stickland et al., 2023), and defining robustness evaluation protocols (Dixon et al., 2018; Morris et al., 2020b). However, previous studies on robustness still exhibit limitations in that they are often focused on specific settings, and thus can hardly generalize to different scenarios, such as new types of data or new model architectures. For example, adversarial training on specific types of noise or attacks might not be generalized to new types (Li et al., 2021b). As the interpretability of neural networks is still a black box for humans, unreliable NLP systems can cause significant social consequences, again highlighting the importance of studying the robustness of NLP models, so that we can build reliable and trustworthy systems in real-world applications that can handle diverse and unpredictable settings.

1.2 Objectives

As discussed above, this thesis focuses on exploring and developing methods to enhance the robustness of NLP systems. Specifically, this thesis aims to address the induced and learned vulnerabilities stemming from **textual noise**, **adversarial attacks**, and **unintended bias**. To achieve these goals, the following research questions are posed.

1. *In real applications, the language variants for textual noise are infinite and new types of noise may continuously emerge. Can augmenting models with additional context signals improve model robustness to noise and perturbations?*

Noise on the Internet is continuously growing, both in variety and quantity. Training with certain types of noise may work temporarily, but could still fail with the creation of new noise types. In this research question, we seek a solution without prior knowledge of

the type of noise to improve the robustness to textual noise and perturbation. To answer this question, we investigate the integration of additional context to alleviate the effect of noise. In Chapter 3, we adopt visual modality and textual error correction to compensate for information loss from noise. Furthermore, in Chapter 4, we introduce conversational context in addition to visual modality to improve robustness. By comparing uni-model vs. multi-modal models, non-contextual vs. contextual models, we can learn the influence of additional context information in improving noise robustness.

2. *Even though adversarial training can mitigate isolated perturbations, iterative attacks that compound successive modifications can be more challenging. Can a denoising or perturbation removal strategy generalize and defend against iterative adversarial perturbations at scale?*

Like the first research question, this research question also focuses on the generalization of robustness to different attacks, but from an adaptive perspective. Adversarial attacks are performed in an iterative style, where the attacker can adjust the attack each time based on the target model’s prediction. To solve this question, in Chapter 5, we explore the possibility of removing noise (i.e., perturbations in the latent space) from the model’s hidden states, which potentially can generalize to all different types of textual perturbation. This question is answered by demonstrating the superior performance of diffusion-based denoising in Chapter 5 against various types of adversarial attacks.

3. *Beyond inference-time defenses, can representation-level disentanglement mitigate unintended biases by separating spurious task correlations during training?*

The first two research questions focus on induced vulnerabilities, while learned vulnerabilities originate from training. This research question shifts the focus from handling or fixing non-standard inputs at test time to improving the modeling at training time. In Chapter 6, we experiment with a multi-task adversarial learning approach to disentangle task correlations and debias the toxicity model. To answer this question, we analyze various spurious correlations and evaluate the model’s performance on unbiased challenge sets.

1.3 Contributions

In this thesis, we made several contributions to understand, analyze and improve the robustness of NLP models to both induced and learned vulnerabilities. Specifically, we showed an improvement on the model’s robustness to textual noise, adversarial attacks, and data bias. Figure 1.1 presents the overall contributions of this thesis:

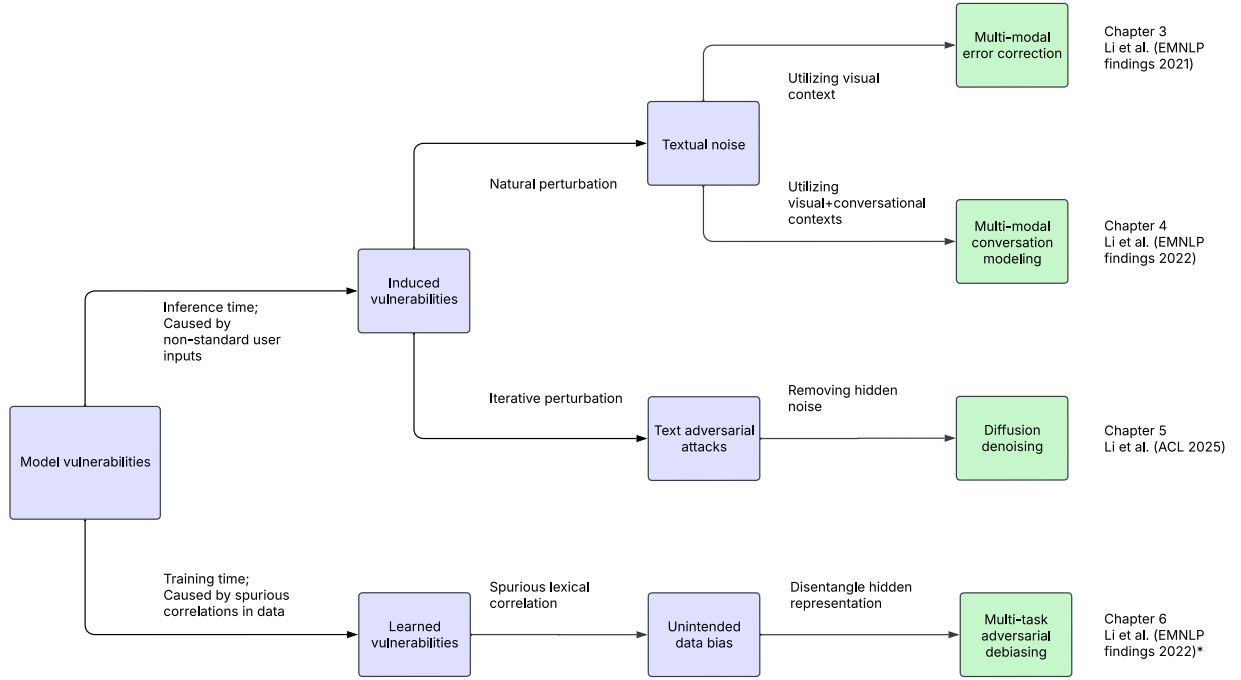


Figure 1.1: Overall contributions of this thesis.

Multimodal error correction training for robust translation: In Chapter 3, we address the robustness issue that arises from textual noise in machine translation models by incorporating visual context. Furthermore, we introduce an error correction training method to advance the model’s robustness against noisy texts. Our findings reveal the effect of multimodality in mitigating the issue of textual noise and show that multimodal models are more robust in noisier settings. We also show that error correction training as an auxiliary task can complement visual context when handling textual noise, and combining both techniques could further improve translation robustness.

Robust multimodal conversational modeling: In Chapter 4, we extend the scope of addressing noise robustness to multimodal conversational modeling. We propose a new task,

multimodal conversation derailment detection, and create a new dataset for this challenging task (§ 4.3). Furthermore, we propose a multimodal hierarchical model, MMConv, which combines textual, visual, and conversational contexts to predict conversation derailment (§ 4.4). Through systematic analysis, we show that conversations provide useful context to improve the accuracy of derailment detection, as well as the performance under noisy texts.

Flexible adversarial defense method for text classification: In Chapter 5, we address the problem of adversarial attacks and propose a novel and flexible method, DiffuseDef, to improve adversarial robustness. The proposed method trains a diffusion layer on top of existing models to perform denoising iteratively on the encoder hidden representations, resulting in an ensembled and denoised hidden representation. By removing latent noise from adversarial attacks, our method achieves state-of-the-art performance against three common black-box and two white-box adversarial attacks. In addition, our proposed method is also flexible and can be applied to any neural network-based model for enhanced adversarial robustness.

Debiased toxicity detection: In Chapter 6, we address the issue of unintended bias in the toxicity detection task. We propose an adversarial multi-task model that extracts representations that share relevant information across tasks, and relies on adversarial learning to disentangle representations that are task-specific and could lead to biases. We experiment on three potential biases in toxicity detection: sentiment, profanity, and identity mention, and show that our proposed method mitigates unintended bias and improves toxicity detection, resulting in a debiased and robust toxicity detection model.

1.4 Publications

1.4.1 Related Publications

The following papers form the core of this thesis and are accepted in peer-reviewed NLP conferences:

1. **Zhenhao Li**, Marek Rei, Lucia Specia. Visual Cues and Error Correction for Translation Robustness. In *Findings of the Association for Computational Linguistics: EMNLP 2021*
2. **Zhenhao Li**, Marek Rei, Lucia Specia. Multimodal conversation modeling for topic derailment detection. In *Findings of the Association for Computational Linguistics: EMNLP 2022*
3. **Zhenhao Li**, Huichi Zhou, Marek Rei, Lucia Specia. DiffuseDef: Improved Robustness to Adversarial Attacks via Iterative Denoising. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (ACL 2025)*.
4. Note that Chapter 6 is based on the paper “Adversarial Multi-task Training for Debaised Toxicity Detection” by **Zhenhao Li**, Marina Fomicheva, Ozan Caglayan, Lucia Specia, which was accepted in *Findings of the Association for Computational Linguistics: EMNLP 2022*, but was withdrawn from publication so that it could first be protected by a patent. It was published as a patent in 2024 “Multi-task neural network for toxicity detection” [US20230419117A1], the copyright of which is held by Epic Games, Inc.

1.4.2 Other Publications

The following publications were completed during the period of this PhD degree. They are related to the topic of this thesis and explore robustness in different NLP systems, but are not discussed in this thesis:

5. Lucia Specia, **Zhenhao Li**, Juan Pino, Vishrav Chaudhary, Francisco Guzmán, Graham Neubig, Nadir Durrani, Yonatan Belinkov, Philipp Koehn, Hassan Sajjad, Paul Michel and Xian Li. Findings of the WMT 2020 Shared Task on Machine Translation Robustness. In *Proceedings of the Fifth Conference on Machine Translation, 2020*.
6. Khetam Al Sharou*, **Zhenhao Li*** and Lucia Specia. Towards a Better Understanding of Noise in Natural Language Processing. In *Proceedings of the International Conference on Recent Advances in Natural Language Processing (RANLP 2021)*

7. Kaustubh D Dhole, Varun Gangal, Sebastian Gehrmann, Aadesh Gupta, **Zhenhao Li**, Saad Mahamood, Abinaya Mahendiran, Simon Mille, Ashish Shrivastava, Samson Tan, Tongshuang Wu, Jascha Sohl-Dickstein, Jinho D Choi, Eduard Hovy, Ondrej Dusek, Sebastian Ruder, and others². NL-Augmenter: A Framework for Task-Sensitive Natural Language Augmentation. In *Northern European Journal of Language Technology (NE-JLT)*, Volume 9, 2023.
8. Huichi Zhou, Kin-Hei Lee, Zhonghao Zhan, Yue Chen, **Zhenhao Li**, Zhaoyang Wang, Hamed Haddadi, Emine Yilmaz. TrustRAG: Enhancing Robustness and Trustworthiness in Retrieval-Augmented Generation, *under review* by the time of thesis submission.

Finally, while not directly related to the thesis, the following publications have been completed during the duration of the PhD degree:

9. **Zhenhao Li**, Marina Fomicheva, Lucia Specia. Exploring Model Consensus to Generate Translation Paraphrases. In *Proceedings of the Fourth Workshop on Neural Generation and Translation (WNGT 2020)*.
10. Lucia Specia, Frédéric Blain, Marina Fomicheva, Chrysoula Zerva, **Zhenhao Li**, Vishrav Chaudhary, André F. T. Martins. Findings of the WMT 2021 Shared Task on Quality Estimation. In *Proceedings of the Sixth Conference on Machine Translation (WMT)*, 2021.
11. Genze Jiang, **Zhenhao Li**, Lucia Specia. ICL’s Submission to the WMT21 Critical Error Detection Shared Task. In *Proceedings of the Sixth Conference on Machine Translation (WMT)*, 2021.

1.4.3 Other Research Contributions

Furthermore, this subsection includes other research contributions conducted during the period of this PhD degree:

²This publication has over 100 co-authors. For brevity, only a subset of the authors is listed here

- Won tied first place on En-Pt translation (Li et al., 2020b) in the 2020 Duolingo Shared Task on Simultaneous Translation and Paraphrase for Language Education (STAPLE).
- Co-organized the WMT 2020 Shared Task on Machine Translation Robustness (Specia et al., 2020a), and presented the findings of the shared task in EMNLP 2020.
- Co-organized WMT 2021 Shared Task on Quality Estimation (Specia et al., 2021).
- Contributed as one of the core developers to an open source software, NL-Augmenter (Dhole et al., 2023), for text transformation and data enhancement.

Chapter 2

Background

In this chapter, we provide a literature review and summarize the background knowledge of key deep learning models used in this thesis and the main robustness issues in modern Natural Language Processing systems that this thesis aims to address. Firstly, in Section 2.1, we introduce the background knowledge for deep neural architectures used in this thesis (§3 - §6), including transformer, BERT-based pre-trained language models, and diffusion models. Then in Section 2.2, we summarize the robustness issues to be addressed in modern NLP systems, i.e. vulnerabilities caused by textual noise, adversarial attack, and unintended bias, etc.

2.1 Deep Learning Models

Deep learning has significantly advanced Natural Language Processing (NLP) by enabling models to learn complex patterns and representations from large amounts of data. Such advancements have led to breakthroughs in various NLP tasks such as machine translation (Sutskever et al., 2014; Bahdanau et al., 2015), sentiment analysis (Kim, 2014), and question answering (Rajpurkar et al., 2016), etc. In this section, we will review the state-of-the-art deep learning models that are used as base models in the subsequent chapters of this thesis. First, the **transformer** architecture (§ 2.1.1) is the backbone of the sequence-to-sequence models used for machine translation in Chapter 3. Secondly, **pre-trained language models** such as BERT

and RoBERTa (§ 2.1.2) are frequently adopted as text encoders across the remaining chapters, including the hierarchical conversational model (MMConv) in Chapter 4, the attack-robust classifier (DiffuseDef) in Chapter 5, and the bias-robust classifier (MTL-adv) in Chapter 6. Finally, the denoising diffusion probabilistic model (DDPM) (§ 2.1.3) is introduced, as it forms the core of the diffusion training and denoising in Chapter 5.

2.1.1 Transformer

The Transformer architecture, introduced by Vaswani et al. (2017), fundamentally changed the landscape of NLP by replacing recurrent structures and instead relying entirely on a self-attention mechanism. Before the introduction of the transformer model, NLP models relied mainly on recurrent structures such as long short-term memory network (LSTM) (Hochreiter and Schmidhuber, 1997a), gated recurrent unit (GRU) (Cho et al., 2014b), etc. These Models process sequences by maintaining each token’s hidden states to model the input sequence’s dependencies and contexts. The hidden state, which is a fixed-length vector, is computed at each timestep based on the previous hidden states and the current input, enabling the model to learn contextual information. However, a fatal limitation of RNN-based models is the loss of information when handling long sequences due to the nature of compressing all the information into a fixed vector. To cope with this issue, Bahdanau et al. (2015) propose attention that dynamically focuses on relevant parts of input tokens during the generation. Based on the success of attention, the transformer model adopts an entirely attention-based architecture and eliminates the recurrence structure for better efficiency and performance.

Transformer is an encoder-decoder model in which the encoder and decoder consist of several identical transformer layers. The encoder layers encode the input sequence with a sequence of vectors, each corresponding to the token representation in the input sequence. As shown in Figure 2.1, the encoded representation is passed to the decoder, which decodes the encoded representation to generate the output in an auto-regressive manner.

In a transformer layer, there are two main components: multi-head attention (MHA) and position-wise feed-forward network. Multi-head attention, which is the key characteristic of

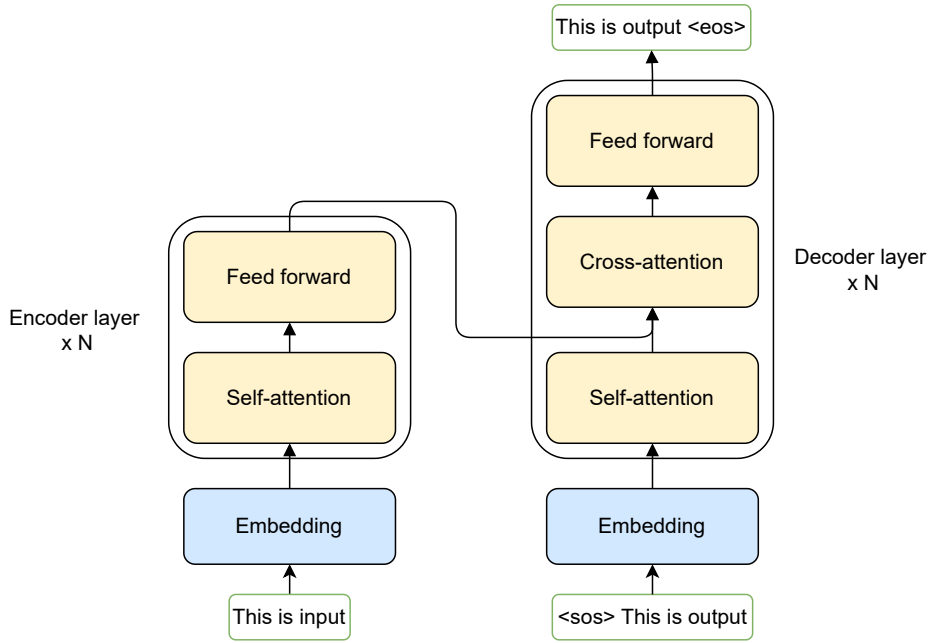


Figure 2.1: Architecture of transformer model. "<sos>" and "<eos>" refer to start-of-sequence and end-of-sequence tokens.

the transformer model, allows the model to weigh the relevance of different tokens and capture contextual relationships. As illustrated in Figure 2.2, each attention head computes a scaled dot-product attention score, which is defined as

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^\top}{\sqrt{d_k}}\right) V$$

where Q , K , and V represent the query, key, and value matrices projected from the sequence representation of the previous transformer layer, and d_k is the dimension of the key vectors used to help stabilize the gradients during training.

The multiple attention heads are created with different query, key, and value projection matrices so that the computation over all heads can be run in parallel for better efficiency.

$$\text{head}_i = \text{Attention}(QW_i^Q, KW_i^K, VW_i^V)$$

where i is the index of the head. The output of all heads is concatenated and projected to the

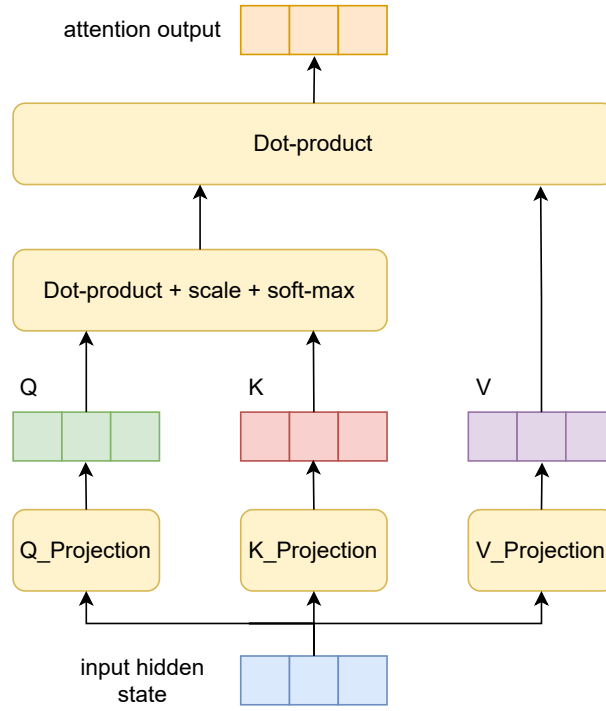


Figure 2.2: Scaled dot-product attention.

same dimension as the input of the MHA layer.

$$\text{MHA}(Q, K, V) = \text{Concat}(\text{head}_1, \text{head}_2, \dots, \text{head}_h) W^O$$

In the encoder transformer layers, the Q , K , and V matrices are all projected from the representation of the input sequence, which is called self-attention. However, in the decoder transformer layer, the Q matrix is from the output of the previous decoder layer, while the K and V matrices are from the encoder output. This mechanism is called cross-attention, and it allows the decoder to dynamically attend to different input tokens during generation.

To help training convergence, each sublayer (i.e., MHA layer and feed-forward layer) in the transformer block also includes a residual connection (He et al., 2015) followed by layer normalization (Ba et al., 2016):

$$\text{LayerNorm}(x + \text{Sublayer}(x))$$

The residual connections help mitigate the vanishing gradient problem, while layer normaliza-

tion can help stabilize training.

Since the transformer removes the recurrent structure, it does not encode the order information in the input sequence. To overcome this issue, a positional embedding is added to the token embedding, taking into account the relative and absolute positions of tokens in the sequence. The positional embedding can either be learned during training or using sine and cosine functions of different frequencies:

$$\begin{aligned} \text{PE}_{(pos, 2j)} &= \sin\left(\frac{pos}{10000^{\frac{2j}{d_{model}}}}\right) \\ \text{PE}_{(pos, 2j+1)} &= \cos\left(\frac{pos}{10000^{\frac{2j}{d_{model}}}}\right) \end{aligned}$$

where pos is the position of the token in the sequence, j is the embedding dimension index, and d_{model} is the dimension of the model.

Due to its big improvement over RNN-based models, transformer has significantly influenced various fields in NLP, CV, etc. In NLP, the transformer has enabled the development of powerful pre-trained models like BERT (Devlin et al., 2019) and GPT (Radford and Narasimhan, 2018), and achieves state-of-the-art performance in tasks such as machine translation (Kocmi et al., 2023), text summarization (Cheng et al., 2023), sentiment analysis (He et al., 2023), etc. In CV, the vision transformer (Dosovitskiy et al., 2021) is applied for image classification, demonstrating its strength and effectiveness in visual tasks. In audio tasks, transformers are also widely used in speech recognition systems, improving the accuracy of transcriptions by leveraging their capacity to model temporal dependencies in audio signals (Dong et al., 2018). Finally, large language models (LLM), which harness the power of massive pre-training for general and multi-modal AI, are mostly based on transformer architecture (Touvron et al., 2023; OpenAI et al., 2024; Anthropic, 2024). These applications highlight the fundamental role of the transformer model in multiple areas, serving as a cornerstone of modern deep learning architectures.

In this thesis, transformer is frequently used as the core network in the next chapters, including the application of transformer as a sequence-to-sequence model for machine translation (§ 3.3.1), and modular usage of the self-attention layer (§ 4.4) and transformer block (§ 5.3.1).

2.1.2 BERT and Other Pre-trained Language Models

While the encoder-decoder transformer was initially proposed in sequence-to-sequence generation tasks such as machine translation, a natural approach to utilize a transformer network for natural language understanding is to use an encoder-only transformer. The BERT model (Devlin et al., 2019), which stands for bidirectional encoder representations from transformers, has been proposed. The BERT model has become a milestone in NLP since its introduction, achieving near-human performance in natural language understanding, and has started the paradigm of pre-training and fine-tuning in NLP.

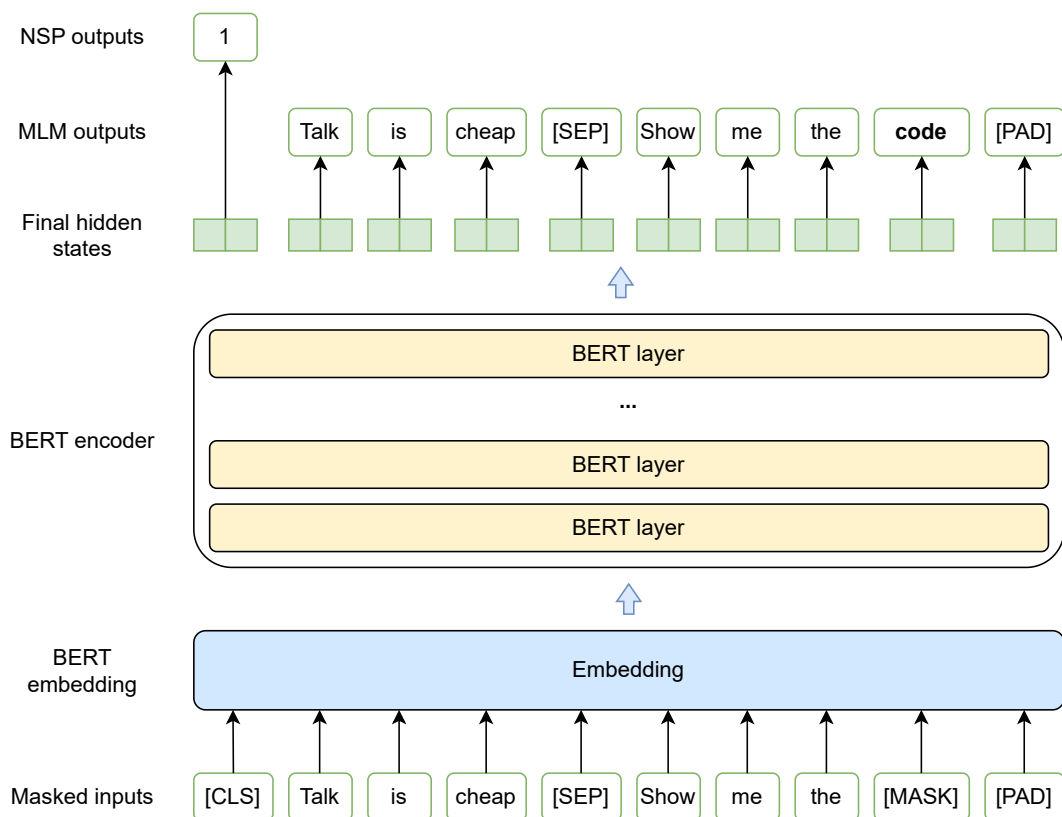


Figure 2.3: Pre-training of BERT.

BERT is pre-trained on massive corpora, i.e., English Wikipedia and BookCorpus (Zhu et al., 2015), enabling it to learn language patterns from massive amounts of data. The pre-trained knowledge can be transferred to downstream tasks such as sentiment analysis, question answering, etc., via fine-tuning, i.e., training the pre-trained checkpoint on domain-specific or task-specific data. BERT employs the self-supervised learning technique during its pre-training

stage (Figure 2.3), with two training objectives during its pre-training:

- Masked language modeling (MLM): In MLM, a portion of the input tokens is randomly masked and the model is trained to predict these masked tokens based on their context. This task optimizes the model to learn the contextual dependencies in the input tokens, leading to a high-quality bidirectional representation from BERT.
- Next sentence prediction (NSP): In NSP, the model learns to understand the relationship between sentences by predicting whether two given sentences follow each other in the original text. This task helps improve performance on tasks requiring sentence-level understanding, such as question answering and sentence similarity.

During the fine-tuning stage, the pre-trained encoder is used as a feature extractor, with an output head added to generate output for the downstream task. Fine-tuning optimizes both the pre-trained encoder and the output head with downstream task data, and the general pre-trained model is thus transformed into a task-specific model. Since the model has developed its language understanding capability during the pre-training stage, fine-tuning requires much less data and computational resources than training a model from scratch. With this benefit, the BERT model has been fine-tuned on various natural language understanding tasks, i.e., the GLUE benchmark (Wang et al., 2018), and achieved near-human performance on reading comprehension, sentiment analysis, etc.

Since the introduction of BERT, several transformer-based models have been developed to address its limitations and improve its performance. RoBERTa (Liu et al., 2019a) improves over BERT by removing the NSP training task and using more pre-training data for better performance. ALBERT (Lan et al., 2020) is a lightweight BERT with factorized embedding parameterization and cross-layer parameter sharing for greater efficiency. ELECTRA (Clark et al., 2020) adopts replaced token detection as the pre-training task. Instead of training with MLM, ELECTRA replaces tokens with plausible alternatives and trains the model to distinguish between the original and replaced tokens. XLNet (Yang et al., 2019b) employs a permutation-based training objective to capture the bidirectional context. In addition to

pre-trained language models with transformer encoders, decoder-based PLMs, such as GPT (Radford and Narasimhan, 2018), are trained autoregressively by predicting the next token given previous tokens. In addition, some models adopt the encoder-decoder framework and pre-train with both natural language understanding and generation tasks, for instance, T5 (Raffel et al., 2020), BART (Lewis et al., 2020), MASS (Song et al., 2019), etc. The family of pre-trained language models like BERT and its successors popularize transfer learning in NLP and establish the new paradigm of pre-training on massive data, followed by task-specific fine-tuning.

In this thesis, we widely adopt different pre-trained language models, including BERT, RoBERTa, XLNet, and T5, in the experimentation. Using pre-trained model weights, we can fine-tune models on various downstream NLP tasks, such as derailment detection (§ 4), topic classification (§ 5), and toxicity detection (§ 6).

2.1.3 Denoising Diffusion Probabilistic Model

The denoising diffusion probabilistic model (DDPM) (Ho et al., 2020) is a generative model introduced to generate realistic high-quality samples in image generation tasks. Traditional generative models such as the generative adversarial network (GAN) (Goodfellow et al., 2014b) and the variational autoencoder (VAE) (Kingma and Welling, 2022) have been successful in image generation; however, they are still limited. For example, GAN is difficult to train due to instability and mode collapse; thus, the model may fail to cover the diversity of the data distribution. VAE, though more stable to train, can generate samples with lower fidelity. Overcoming these limitations, DDPM leverages a diffusion process to model data distributions. During training, the data sample is added with random noise over several steps and eventually converted to pure noise. The model is trained to reverse this diffusion process step by step, with each step removing the added noise; therefore, at inference time it can effectively reconstruct the data from sampled noise.

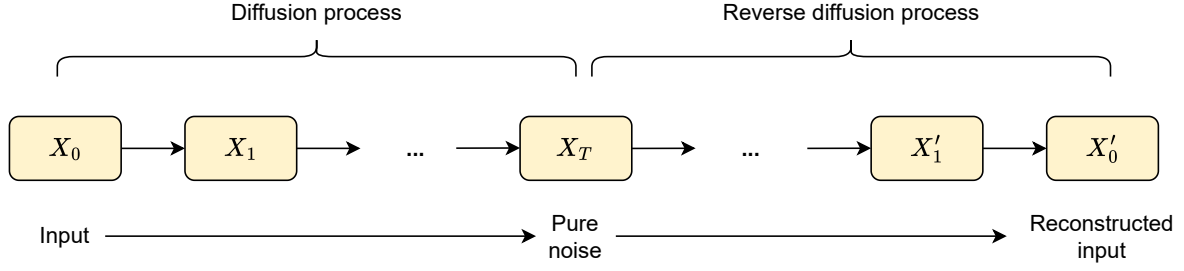


Figure 2.4: Diffusion and reverse diffusion process in DDPM.

Diffusion process At training time, the diffusion process adds random noise sampled from a Gaussian distribution to the data x_0 over T timesteps, resulting in a sequence of noisy variables $x_1, x_2, \dots, x_T$. This forward diffusion process at step t can be seen as a conditional probability $q(x_t|x_{t-1}) = \mathcal{N}(x_t; \sqrt{1 - \beta_t} \cdot x_{t-1}, \sqrt{\beta_t} \cdot I)$:

$$x_t = \sqrt{1 - \beta_t} x_{t-1} + \sqrt{\beta_t} \epsilon \quad (2.1)$$

where $\epsilon \in \mathcal{N}(0, I)$ is the sampled noise, and β is a variance schedule that controls the amount of noise added at each step where $0 < \beta_1 < \beta_2 < \dots < \beta_T < 1$.

Using the re-parameterization trick from VAE (Kingma and Welling, 2022), x_t can be sampled directly with any arbitrary noise conditioned on x_0 :

$$q(x_t|x_0) = \mathcal{N}(x_t; \sqrt{\bar{\alpha}_t} \cdot x_0, (1 - \bar{\alpha}_t) \cdot I) \quad (2.2)$$

where $\alpha_t = 1 - \beta_t$ and $1 - \bar{\alpha}_t = \prod_{i=1}^t \alpha_i$. With this re-parameterization trick, the noise can be sampled to construct x_t from x_0 efficiently, without the need to iteratively add noise. Therefore, during training, the neural network is optimized to predict the added noise at time t , and therefore the training objective is to minimize the mean square error between the sampled noise ϵ and the predicted noise $\epsilon_\theta(x_t, t)$:

$$L_t = \mathbb{E}_{t, x_0, \epsilon_t} [\|\epsilon - \epsilon_\theta(x_t, t)\|^2] \quad (2.3)$$

Reverse diffusion process Since the model is trained to predict the added noise in the forward diffusion step, it can be used to remove the noise at the inference time by sampling from $q(x_{t-1}|x_t)$. This distribution is approximated by the model and a reverse diffusion step can be defined as:

$$p_\theta(x_{t-1}|x_t) = \mathcal{N}(x_{t-1}; \mu_\theta(x_t, t), \Sigma_\theta(x_t, t)) \quad (2.4)$$

Therefore, at inference time the $x_T \sim \mathcal{N}(0, I)$, which is pure noise, is first sampled from a Gaussian distribution. Then an iterative process of removing model-predicted noise ϵ_θ is performed to produce the denoised sample from x_{T-1} to x_0 .

$$x_{t-1} = \frac{1}{\sqrt{\alpha_t}} \left(x_t - \frac{1 - \alpha_t}{\sqrt{1 - \bar{\alpha}_t}} \epsilon_\theta(x_t, t) \right) + \sigma_t z \quad (2.5)$$

where $z \in \mathcal{N}(0, I)$, and $\sigma_t = \sqrt{\beta_t}$.

Similar to the role of BERT in natural language understanding, DDPM also significantly contributed to the field of image generation and text-to-image synthesis. The denoising diffusion implicit model (DDIM) (Song et al., 2021) provides a deterministic version of DDPM, allowing faster sampling by reducing the number of steps required. Guided diffusion model (Dhariwal and Nichol, 2021) introduces guidance into DDPM by guiding the diffusion process towards an arbitrary class label through a pre-trained classifier. Extending the idea of diffusion training, models like DALL-E (Ramesh et al., 2021) and Stable Diffusion (Rombach et al., 2021) leverage diffusion processes to produce detailed and coherent images from textual descriptions.

Although the diffusion model originates in computer vision, in Chapter 5, we extend its application to NLP models to improve the robustness to adversarial attack. The strategy of iteratively predicting and subtracting noise from hidden states forms the core logic of our work to denoise perturbations from adversarial attacks.

2.2 Robustness Issues in NLP

2.2.1 Handling Textual Noise

Textual noise, such as misspellings, grammatical errors, code-switching, obfuscation, and leet speak, poses significant challenges for Natural Language Processing (NLP) models. Robustness to such noise is crucial for deploying NLP systems in real-world applications, where clean, standard text cannot be guaranteed. While pre-trained language models (PLMs) like BERT (Devlin et al., 2019), GPT (Radford and Narasimhan, 2018), and T5 (Raffel et al., 2020) have achieved remarkable success on standard benchmarks, they often struggle with noisy input, highlighting a gap between benchmark performance and real-world robustness. These pre-trained language models are typically trained on large, clean corpora (e.g., Wikipedia, news articles) without significant exposure to noisy data. Consequently, the models exhibit sensitivity to textual perturbations, such as misspellings and syntactic variations.

Recent research demonstrates that even minor textual noise, such as character swaps or typos, can lead to severe performance degradation across various NLP tasks. This fragility underscores the reliance of PLMs on token-level embeddings, which are disrupted by even small perturbations in the input. Neerudu et al. (2023) evaluate the robustness of PLMs against different types of perturbation on classification tasks (i.e., the GLUE benchmark) and show a performance drop on all tasks. Balakrishnan et al. (2022) perform robustness evaluation on sentiment analysis task and show that prefix-tuning is significantly brittle to noise, with a decrease in performance on most corrupted datasets of increasing noise levels. Cooper Stickland et al. (2023) expand the exploration of textual noise in multilingual settings and show that the harm of textual noise can be language-specific. Likewise, researchers have studied the impact of noisy input to pre-trained models in various NLP tasks, e.g., machine translation (Belinkov and Bisk, 2018; Khayrallah and Koehn, 2018; Heigold et al., 2018; Michel and Neubig, 2018), reading comprehension (Huang et al., 2020; Náplava et al., 2021), question answering (Peskov et al., 2019), sentiment analysis (Kumar et al., 2020), etc. Addressing the issue of noisy input in NLP has therefore drawn attention in NLP research, especially with the widespread avail-

ability of data from social media platforms such as Twitter/X, Reddit, among others, creating not only new opportunities, but also different challenges (Karpukhin et al., 2019).

2.2.1.1 Categorization of Textual Noise

To deal with the robustness issue caused by textual noise, it is important first to understand the categorizations of noise specific to NLP. Unlike in audio or image processing, where noise typically refers to measurable perturbations such as background sounds, pixel-level corruption, or compression artifacts, the concept of noise in text is more nuanced and less clearly defined. Noise in text manifests itself as deviations from linguistic norms, which are inherently context-dependent and multifaceted. For instance, a misspelling, grammatical error, or informal abbreviation may not always be considered noise, as its impact depends on the task and intended meaning. In addition, unlike the continuous nature of audio and image signals, text is discrete, which means even minor perturbations – such as a single swapped character – can drastically alter the meaning or disrupt tokenization.

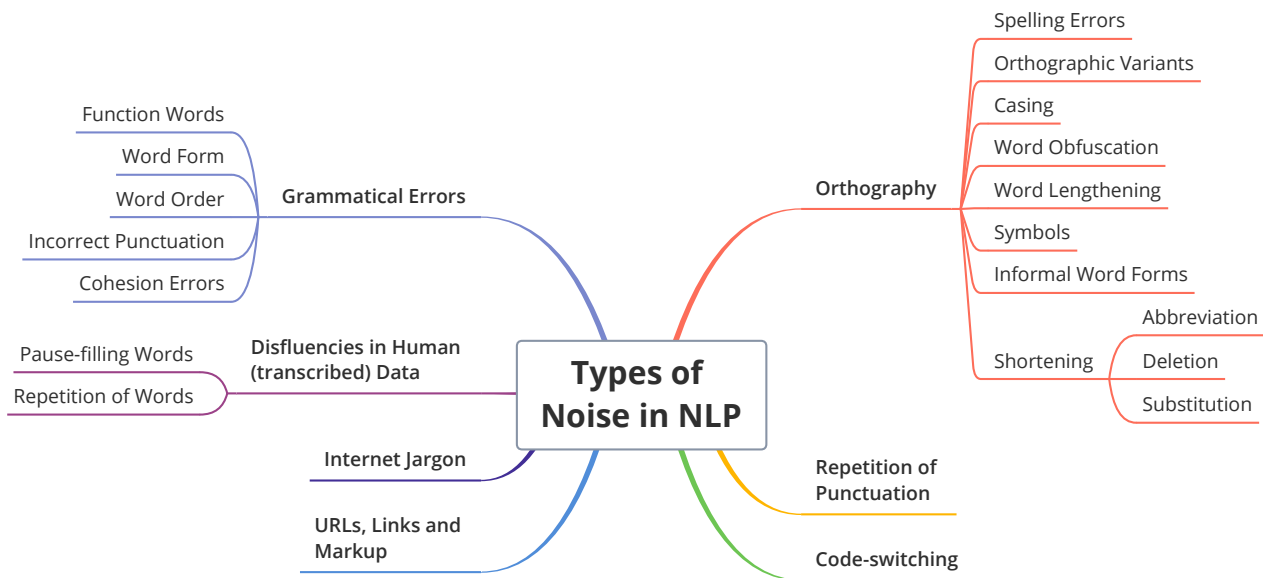


Figure 2.5: Categorizations of noise in NLP (Al Sharou et al., 2021).

Based on such understanding, an intuitive way to categorize the types of noise in NLP is to distinguish their characteristics and potential impact on downstream applications. Figure 2.5 presents a high-level overview of different types of noise in NLP:

- **Orthography:** this type of noise is concerned with the way words are written, including errors like spelling errors and variations or deviations from the standard way of writing, *e.g.* word obfuscation (like “gr8”), word lengthening (such as “pleeeease”), etc.
- **Grammatical errors:** This type of noise implies a deviation from the grammatical rules of a language apart from spelling errors (Burchardt, 2013; Garnier and Saint-Dizier, 2016), such as incorrect use of function words, word orders, punctuation, etc.
- **Disfluencies in Human (transcribed) Data:** This type covers disfluencies that occur in spontaneous spoken language (Shriberg, 1994), including pause-filling words like “uh”, “er”, “um” which generally do not carry any meaning, and repetition of words during speech.
- **Repetition of Punctuation:** Unlike grammatical misuse of punctuation, this type of noise refers to instances where punctuation marks such as exclamation mark or question mark are repeated to serve a purpose, *i.e.* for emphasis, *e.g.* “Really?????”
- **Code-switching:** This type of noise refers to the alternation between different languages in a single sample, for instance “努力ing”.
- **Internet Jargon:** This type refers to new words and acronyms that have gained special meaning and usage on certain social media platforms. Words such as “downvote”, “upvote”, and acronyms such as “TIL” for “Today I Learned” are examples of frequent jargon on Reddit.
- **URLs, Links, and Markup:** This type includes web addresses and hyperlinks, *e.g.* HTML tags, URLs, and Hashtags, and these elements might provide additional contextual information. For example, the “@mentions” on Twitter (*a.k.a.* X) carries the information of the account profile (Gorrell et al., 2015).

The categorization provides a high-level differentiation of various types of noise in NLP, while it cannot be exhaustive as textual noise in real-world applications is constantly evolving, and new forms of noise may emerge with changes in language usage.

2.2.1.2 Improving Robustness to Noise

Existing techniques dealing with noise fall into two main perspectives: data-based approaches and model-based approaches. The former focuses on modifying the input data (either training data or test time data) to make the models more robust to noise, while the latter looks into modifications in model architecture or training objectives to enhance the intrinsic robustness of a model to noise.

Data-based approaches An intuitive method to improve models' performance on noisy text is to train deep learning models on noisy data, a.k.a adversarial training. Noisy data can be obtained from real-world texts (Michel and Neubig, 2018), or via synthetic data augmentation (Belinkov and Bisk, 2018; Náplava et al., 2021; Li and Specia, 2019). Given that real-world noisy texts might not be as easy to obtain as clean texts in specific tasks, synthetic data augmentation is preferred. Data augmentation involves creating noisy variants of the original clean dataset by introducing perturbations such as typos, grammatical errors, obfuscations, or paraphrasing. This exposes models to diverse noise patterns, improving their generalization capabilities. However, this technique may increase the size of the training data, thus requiring more computational resources.

In addition, the other direction is to perform normalization to test-time inputs, which focuses on cleaning or transforming noisy input texts into standard forms before NLP models process them. This technique may involve simple pre-processing (Al Sharou et al., 2021), noise removal with pre-trained language models (Sun and Jiang, 2019; Sengupta et al., 2021), paraphrasing with large language models (Bolding et al., 2023; Zhang et al., 2024), etc. By normalizing text at inference time, the test input would better align with the training distribution, improving overall model performance on noisy data without requiring extensive retraining on augmented datasets. However, the main weakness of this approach is the increased inference time introduced by the normalization step.

Model-based approaches Apart from increasing noise in training data or decreasing noise in test data, some existing works focus on addressing the noise problem from a model architecture perspective. A prominent strategy is to use character-level models to minimize out-of-vocabulary (OOV) words caused by perturbation (Lee et al., 2017; Heigold et al., 2018; Malykh, 2019; Kurita et al., 2020; Rosales Núñez et al., 2021). Furthermore, recent research also focuses on byte-level models, which completely eliminate the need for tokenization and thus eliminate OOV words (Yang et al., 2019c; Xue et al., 2022; Nguyen and Chen, 2023). Other types of models apply denoising during training (Lewis et al., 2020; Ji et al., 2024) or regularization to minimize the distance between noisy and clean data (Chen et al., 2023), thus improving the model’s robustness to noise at inference time.

In addition, hybrid approaches combine data-based and model-based strategies to improve the resilience of NLP models to textual noise. Srivastava and Chiang (2023) propose BERTwch, which embeds the BERT encoder between additional encoder layers and continues pre-training the model on synthetic noisy texts to improve robustness to dialectal and noisy texts. Cooper Stickland et al. (2023) propose robust contrastive pre-training by synthesizing noisy data and applying contrastive learning to train the model to distinguish between clean and noisy inputs. The approach is shown to improve robustness to noise in multilingual settings.

Despite advances in improving robustness to textual noise, noise still remains a challenging problem, even in the era of large language models. Recent research studies the robustness of large language models such as GPT-4, and Llama3, and finds that although LLMs exhibit stronger robustness than models like BERT, they are not immune to performance degradation when confronted with noisy texts (Wang et al., 2024; Singh et al., 2024). These findings highlight that the issue of textual noise in NLP is far from resolved, re-emphasizing the need for continued research to develop more robust and reliable NLP systems.

2.2.2 Defending Adversarial Attack

2.2.2.1 Textual Adversarial Attack

The concept of adversarial attack was first proposed in Computer Vision (CV). Szegedy et al. (2013) first showed that images applied with imperceptible noise could fool neural image classifiers and lead to a high prediction error rate. To overcome this problem, an intuitive method is to train the model on adversarial images for better robustness. The Generative Adversarial Network (GAN) (Goodfellow et al., 2014a) is one of the most widely adopted approaches to adversarial training. A GAN consists of a generator and a discriminator. The generator generates adversarial images from sampled noise and tries to fool the discriminator, while the discriminator distinguishes between real and adversarial images. Since images are represented with continuous signals, the gradients in the neural network can be back-propagated through the discriminator to the generator. Therefore, the generator and discriminator could optimize in parallel. With a well-trained generator, the model could benefit from the adversarial images created by the generator and improve its robustness.

However, adversarial attacks in NLP are different from attacks in CV. An important assumption is that changes in adversarial examples should be imperceptible to humans. In the image domain, the degree of imperceptibility is measured with the L_p distance (Eq. 2.6) between the original image x and the adversarial image x' .

$$L(x, x')_p = \sqrt[p]{\sum_{i=1}^n |x_i - x'_i|^p} \quad (2.6)$$

This assumption can be easily fulfilled with images, as a human can hardly distinguish changes in a few pixels. However, small changes in texts can be recognized by a human, even changing a character. Therefore, an adversarial example in NLP is considered valid if it shares the same semantic meaning as the original text. However, the goal of textual adversarial attacks is the same as that of visual adversarial attacks, which is to maximize the likelihood of incorrect predictions by a neural network. Adversarial texts are typically created by introducing subtle

perturbations to the original text, such as character swapping (Gao et al., 2018; Ebrahimi et al., 2018a), synonym-substitutions (Ren et al., 2019; Morris et al., 2020a; Yoo and Qi, 2021), and paraphrasing (Gan and Ng, 2019; Huang and Chang, 2021), etc. The semantic similarity between adversarial and original texts is measured with vector-based metrics such as Euclidean distance and cosine similarity. The original and adversarial texts are first transformed into a vector space using text encoders such as the Universal Sentence Encoder (USE) (Cer et al., 2018), and then the similarity between the two vectors is calculated. In recent research, valid adversarial examples are required to have a similarity score above a threshold.

Textual adversarial attacks can be applied to all NLP tasks as they only perturb the input. Taking the text classification task as an example, given a classifier $\mathcal{C}(\mathbf{x})$ that maps an input sequence of words $\mathbf{x} = [w_1, w_2, \dots, w_L]$ to its designated label y , the goal of the attack model is to construct an adversarial example $\mathbf{x}' = \mathbf{x} + \delta$ to fool the classifier, where δ is a subtle adversarial perturbation constrained by $\|\delta\| < \omega$. The adversarial example $\mathbf{x}'$ is considered a successful attack if it leads to an incorrect prediction $\mathcal{C}(\mathbf{x}') \neq y$. The attacker can iteratively generate multiple adversarial examples and query the classifier to obtain a successful attack, whereas the classifier must consistently return the correct prediction within a specified number of query attempts to be considered robust (Fig 2.6). Adversarial attacks can be categorized into white-box attacks and black-box attacks depending on whether the attacker can access the parameters of the classifier.

Common textual adversarial attack methods adopt a two-stage process to construct effective adversarial examples: *word importance ranking* and *word substitution*. In the first stage, words or subwords are ranked based on their influence on the model’s prediction. This is measured by leveraging either gradient information (Liu et al., 2022) (*white-box*) or changes in prediction probabilities when words are removed (Jin et al., 2020) or masked (Ren et al., 2019; Li et al., 2020a) (*black-box*). In the second stage, candidate words are substituted with synonyms (Zang et al., 2020), perturbed variants (Gao et al., 2018), or outputs from masked language models (Garg and Ramakrishnan, 2020; Li et al., 2020a). The substitution process is guided by various constraints to ensure that the adversarial example remains natural and semantically equivalent to the original text. Despite the current advances of pre-trained language models in NLP,

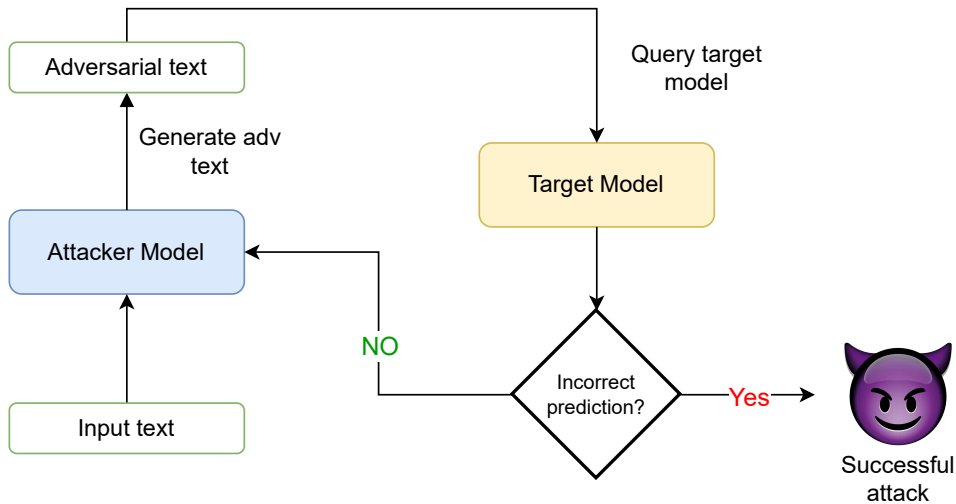


Figure 2.6: Illustration of adversarial attack pipeline.

current textual adversarial attacks still pose significant challenges to NLP models (Liu et al., 2022; Xu et al., 2021b; Yuan et al., 2023), highlighting the necessity for defense methods for better adversarial robustness.

2.2.2.2 Categorization of Adversarial Attacks

Adversarial attacks in NLP can be categorized by different dimensions, such as access to the model, granularity of the attack, and attack objective. Although other criteria may also distinguish different attacks, these dimensions are most commonly discussed in the literature.

Granularity One of the most intuitive ways to classify adversarial attacks is based on the granularity of the perturbation. Character-level attacks introduce minimal edits, such as swapping, inserting, or deleting characters within words (Gao et al., 2018; Ebrahimi et al., 2018a). However, character modifications might introduce typos and thus can be perceptible by humans. Word-level attacks substitute one or more words in the text with synonyms or other valid replacements, aiming to keep the overall meaning (Ren et al., 2019; Morris et al., 2020a; Yoo and Qi, 2021), though they may introduce ungrammatical results. Instead of modifying characters or single words, paragraph-level attacks insert, delete, or paraphrase whole phrases or sentences (Gan and Ng, 2019; Huang and Chang, 2021), thus ensuring semantic preservation

and grammatical correctness. However, the cost of generating an adversarial example can be high because the attack model is required to generate the full text to construct adversarial examples.

Model access Based on the access that attackers have regarding the target model, different adversarial attacks can be divided into *white-box* attacks and *black-box* attacks. Under the white-box settings, attackers have full access to the model’s parameters, architecture, and gradients. By leveraging these signals, attackers can craft adversarial examples more efficiently. Common white-box attacks include Hotflip (Ebrahimi et al., 2018c), GBDA (Guo et al., 2021), CWBA (Liu et al., 2022), etc. However, in real applications, white-box attacks are less realistic, as the target model access can hardly be exposed to the attacker. In contrast, black-box attackers only observe model outputs (e.g., predicted probabilities and logits) and do not have direct access to model parameters or gradients. Instead, attackers iteratively probe the model with candidate adversarial examples until the target model generates an incorrect output (Gao et al., 2018; Li et al., 2019a; Ren et al., 2019; Jin et al., 2020; Garg and Ramakrishnan, 2020; Li et al., 2020a). Since black-box attacks require iterative querying of the target model, the cost of the attack can be much higher than that of the white-box attack.

Attack objective A third categorization considers whether the attack leads to a specific incorrect label (*targeted* attack) or any misclassification results (*untargeted* attack). In text generation tasks, targeted attacks aim to generate specific output phrases, while for text classification tasks, most attacks are untargeted, aiming only to cause the model to fail, regardless of which incorrect label is predicted.

2.2.2.3 Textual Adversarial Defense

To mitigate the performance degradation caused by adversarial attacks, various adversarial defense methods have been developed. Adversarial defense aims to safeguard the performance and reliability of the model under intense adversarial pressure. Common adversarial defense

methods can be classified into four categories: *adversarial training*, *denoising*, *ensembling*, and *robustness certification*.

Adversarial training Similar to tackling noisy data, adversarial training is also the most common approach for dealing with attacks. The adversarial training extends the model’s training process by explicitly introducing adversarial examples in the training set, thereby improving the model’s robustness to test-time perturbations. Unlike generic noise adversarial training, the perturbations for adversarial attack are often targeted, and some may even be samples from successful attacks to make up the model’s vulnerability. The adversarial training examples can be created via synthetic data augmentation (Pruthi et al., 2019) (i.e., injecting noise to clean texts), noise injection to embeddings (Madry et al., 2018; Zhu et al., 2020; Wang et al., 2021; Xi et al., 2022; Gao et al., 2023), and data mixup (Si et al., 2021). While adversarial training can improve the robustness to certain types of attacks, it is often resource-intensive because of the expanded training set. Furthermore, adversarial training also assumes similar train-test adversarial examples and thus tends to overfit specific types of adversarial attacks (Li et al., 2021b).

Denoising Denoising-based methods mitigate adversarial effects by cleaning or normalizing inputs to a standard form before forwarding them to the model for output. The denoising process can be done on either text-level, i.e., rewriting adversarial texts (Zhou et al., 2019), or vector-level, i.e., removing adversarial noise from latent representations (Nguyen Minh and Luu, 2022; Wang et al., 2023; Moon et al., 2023; Yuan et al., 2024). Denoising-based methods introduce extra difficulty for the attacker model in finding vulnerable words to attack, thus improving adversarial robustness (Wang et al., 2023). Nevertheless, denoising might lead to different semantics between clean text and adversarial text, and as a result may hinder clean text performance.

Ensembling Ensembling methods rely on multiple predictions on different variants of the inputs. By leveraging the predictions (e.g., predicted label or logits on each class) on multiple

semantically similar inputs, ensembling-based defenses aim to prevent single point of failure caused by adversarial perturbation (Ye et al., 2020; Zhou et al., 2021c; Zeng et al., 2023; Li et al., 2023). Ensembling is especially effective against black-box attacks due to its attacking nature of iterative querying, but comes with a fatal limitation in inference-time efficiency, as model predictions are needed on every ensemble, increasing the inference time linearly with the number of ensembles.

Robustness certification Certified robustness approaches provide theoretical guarantees that the model prediction remains unchanged within a certain range of textual perturbations. These methods often adapt techniques from interval-bound propagation (IBP) (Jia et al., 2019; Shi et al., 2020; Zhou et al., 2021c) or randomized smoothing (Ye et al., 2020; Zeng et al., 2023). IBP focuses on the worst-case performance by defining small intervals for each token embedding $[e_i - \epsilon, e_i + \epsilon]$ and propagating these intervals forward through the model layers to derive bounds on the final output. Randomized smoothing aims to produce a smooth and averaged classifier that is less sensitive to small perturbations in the input, which is often achieved by adding small noise to the embedding or injecting text perturbations. Although robustness certification provides a strong theoretical basis for adversarial robustness, in real applications, due to the discrete nature of texts, adversarial examples may not always stay within a small radius of the original texts. Therefore, the attack might still escape from the defense and thus cause incorrect outputs.

2.2.3 Mitigating Unintended Bias

Bias in NLP refers to the discriminative outputs of a model that may target a specific person or group. During training, the model can learn associations that are not explicitly patterned but emerge naturally from the data or modeling choices, thus causing unintended bias during inference (Dixon et al., 2018). For example, embedding models may enforce gender bias by associating “man” with “computer programmer” and “woman” with “homemaker” (Bolukbasi et al., 2016); toxicity detection models may predict toxicity when a specific identity group is

present in the input due to lexical correlation in the training data (Garg et al., 2023), etc. The bias in the model can be learned from the data itself that contains societal bias and stereotypes (Blodgett et al., 2020), or might be caused by sampling issues that lead to an underrepresentation of the real data distribution (Sap et al., 2019). The unintended bias in NLP models can perpetuate negative stereotypes and may lead to real-world discrimination and harm to target groups.

To mitigate unintended bias in NLP models, existing research has been conducted mainly in two directions: *data sampling* and *debiased training*. The former approach focuses on improving the training set via strategies such as data balancing (Dixon et al., 2018), and synthetic data augmentation (Park et al., 2018). In contrast, debiased training modifies the model architecture or the training process to reduce biased associations. The most common approach is to integrate regularization in biased terms to minimize bias during training (Zhao et al., 2018; Elazar and Goldberg, 2018; Kennedy et al., 2020a; Kaneko and Bollegala, 2021), however, this approach requires prior knowledge of the biased terms. Recently, interpretability-based approaches have been proposed in which information from gradient (Bose et al., 2022), attention (Attanasio et al., 2022), or model weights (Mamta et al., 2024) is used to debias the model during training.

Recent work has also focused on combining data and modeling approaches to mitigate model learned bias. Karimi Mahabadi et al. (2020) first train a bias-only model and use it to generate predictions to adjust the loss for the debiased model, and thus down-weight the potential bias in the training data. Schick et al. (2021) extend the application to the NL generation task, and utilize the model’s internal knowledge to identify potential bias, followed by prompting to guide the generation and mitigate bias in the output. Du et al. (2023) train several bias models on subsets of the training set to extract bias features, and then use information entropy to purify the biased features. Cheng and Amiri (2024) apply data augmentation to add perturbation and generate biased input, and use contrastive training loss to debias the model training. In summary, mitigating unintended bias in NLP requires a careful combination of both data and modeling improvement. Ensuring representative datasets and balanced sampling can reduce the likelihood of learning biased associations, while debiasing methods within the training pipeline provide crucial modeling robustness that alleviates overfitting on specific biased inputs.

Chapter 3

Multimodal Error Correction for Robustness against Textual Noise

As described in Chapter 2, noise is inherently present in texts on the Internet, e.g., misspelled words and ungrammatical constructions. The presence of such noise can significantly degrade the performance of common NLP systems, with neural machine translation (NMT) being particularly affected. NMT models are sensitive to input noise, and existing robustness techniques tend to fail when faced with unseen types of noise, resulting in degraded performance on clean texts. In this chapter, we address three common types of realistic noise frequently introduced by human errors and propose the incorporation of *visual context* to improve translation robustness in noisy environments. In addition, we introduce a novel *error correction training* regime that can be used as an auxiliary task to further enhance the model’s ability to handle noisy inputs. Our experiments on English-French and English-German translation tasks show that multimodal and error correction approaches improve model robustness to noisy texts while preserving translation quality on clean texts.

This chapter is based on the paper “Visual Cues and Error Correction for Translation Robustness” published in *Findings of EMNLP, 2021* (Li et al., 2021a). Codes and configurations are available at <https://github.com/Nickeilf/Visual-Cues-Error-Correction>.

3.1 Introduction

Neural Machine Translation (NMT) has been shown to be highly sensitive to noise (Belinkov and Bisk, 2018; Michel and Neubig, 2018; Ebrahimi et al., 2018b), with even minor perturbations in the input often leading to mistranslations. To improve the robustness of NMT models, previous research primarily focuses on adapting the model to noisy data through techniques such as fine-tuning on noisy texts (Michel and Neubig, 2018; Alam and Anastasopoulos, 2020), synthetic noise injection (Belinkov and Bisk, 2018; Cheng et al., 2018; Karpukhin et al., 2019), and data augmentation through back-translation (Berard et al., 2019; Vaibhav et al., 2019; Li and Specia, 2019), etc. In these approaches, the translation model is either trained from scratch or fine-tuned on the noisy data to learn noise patterns effectively.

However, these adaptation-based methods assume that the model is trained with noise patterns identical to those in the test data. Since noise in real-world applications varies widely and can be unpredictable, these methods will fail on unseen noise types (Belinkov and Bisk, 2018). To address this limitation, leveraging extra contexts beyond just the noisy input text could offer a more robust solution to handle real-world noise. However, methods that incorporate such extra context for translation robustness have not yet been thoroughly investigated.

Studies in Multimodal Machine Translation (MMT) have demonstrated that visual information improves translation quality, particularly when the textual context is incomplete (Caglayan et al., 2019; Imankulova et al., 2020; Caglayan et al., 2020). However, as exemplified by Caglayan et al. (2019) (Figure 3.1), an MMT model trained on clean data was unable to handle noise. When the word “son” is misspelled as “song”, the model disregards the visual information and uses the literal translation “chanson”. The MMT model attends to the relevant region in the image and generates the intended translation “enfant” only when the noise is masked by a placeholder in the input, imitating an out-of-vocabulary (OOV) word.

Given that the visual modality has been shown to aid in predicting unknown words, we explore whether adding multimodal information to adaptation-based methods would further improve translation robustness. To answer this question, we build MMT models in conjunction with

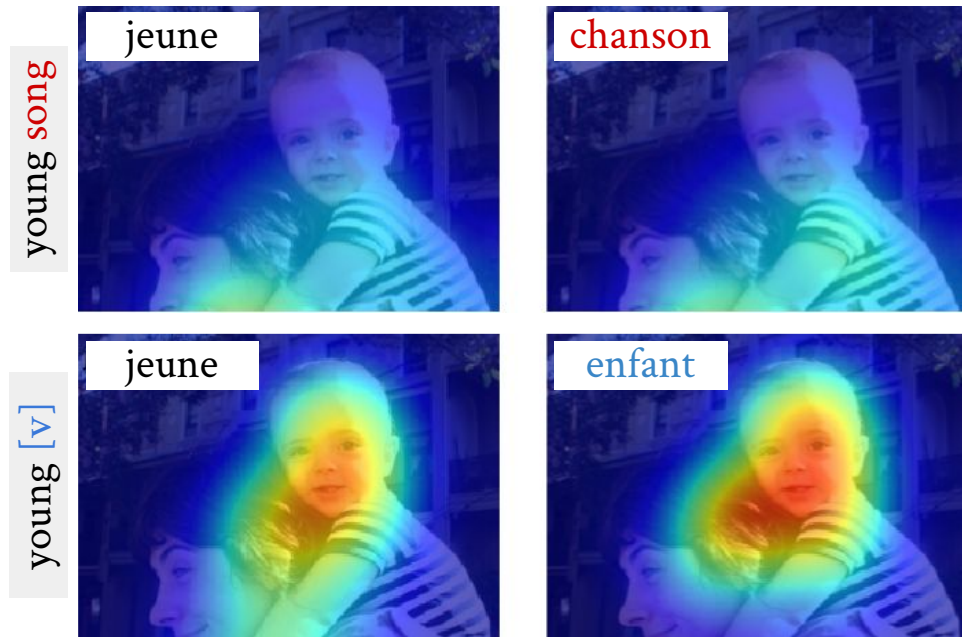


Figure 3.1: As shown by Caglayan et al. (2019), multimodality can help translate unknown words, but fails when there is noise in the input. In this example, the misspelled word “song” is correctly translated as “enfant” (child) when it is replaced with an unknown token, but translated literally as “chanson” (song) otherwise.

noise injection techniques and evaluate the performance during training and inference on both noisy and clean data. To further improve robustness, we extend the current *adversarial training* method (i.e., training NMT models on noisy texts) and propose an *error correction training* method. In addition to training the model with noise-injected source sentences and their clean translation counterparts, we introduce error correction as an auxiliary task and add a separate decoder to the model, which is used to denoise the source sentence.

Our main contributions can be summarized as follows.

- To the best of our knowledge, this is the first work combining adversarial training with *multimodal MT* to improve translation robustness. We evaluate the robustness on three types of noise that simulate common human errors. Systematic experiments reveal that multimodality can improve model performance on both known and unseen noise.
- We propose an *error correction training* method for translation by introducing denoising as an auxiliary task. We show that the robustness of both the NMT and MMT models is improved with this method.

- We demonstrate that the model using visual features also learns to correct grammatical errors more accurately, indicating the potential for multimodal monolingual error correction.

The remainder of this chapter is organized as follows. Section 3.2 provides an overview of related work in unimodal and multimodal machine translation. Section 3.3 details the methodology, including the types of noise injected and the error correction training method. In Section 3.4, we outline the experiment setup, including datasets and model configurations. We present the experiment results in Section 3.5, and further analysis in Section 3.6. Section 3.7 provides a qualitative evaluation with attention map visualization and generation examples. Finally, Section 3.8 concludes the chapter and discusses potential directions for future research.

3.2 Related Work

Robust NMT As discussed in Chapter 2, even though NMT models can achieve high performance on clean data, they are very brittle to non-standard inputs, such as noisy texts (Belinkov and Bisk, 2018). Different types of noisy data have been proposed to test translation robustness, e.g., synthetic word perturbations (Belinkov and Bisk, 2018), grammatical errors (Anastasopoulos et al., 2019), and user-generated texts from social platforms (Michel and Neubig, 2018; Li et al., 2019b; Specia et al., 2020b).

The most straightforward approach to improve translation robustness is to train the model on noisy data, which is referred to as adversarial training. Since parallel data with noisy source sentences and clean translations is difficult to obtain, the clean training data is often injected with different types of artificial noise, e.g., random word perturbations like character insertion, deletion, and substitution (Belinkov and Bisk, 2018; Karpukhin et al., 2019; Passban et al., 2020; Xu et al., 2021a), noise generated via back-translation (Berard et al., 2019; Vaibhav et al., 2019; Li and Specia, 2019), and adversarial examples generated by generative model (Cheng et al., 2018, 2019, 2020). Although this method has been shown to improve NMT performance on noisy data, the types of noise used so far are not common in real data. For example, it

would be highly unlikely for human authors to misspell the word “robust” as “zobust”, but such random transformations are used when synthesizing noisy training data for MT. In addition, back-translation paraphrases the texts to introduce noise, however, such noise is less realistic than human-generated errors, which include realistic misspellings and grammatical errors in daily typing. In adversarial approaches for other NLP tasks, Ribeiro et al. (2020) and Ma (2019) introduce various methods to inject both artificial and realistic noise. Inspired by their work, we focus on three types of noise that are commonly generated by humans in real texts and experiment with these for the translation task.

MMT Multimodal machine translation extends the framework of NMT by incorporating extra modalities, e.g., image (Specia et al., 2016a), audio (Sulubacak et al., 2020), etc. In our case, the extra modality is given as visual features from an image network to complement the textual context. In standard MMT, these features can be fused with textual representation by simple operations such as concatenation (Caglayan et al., 2016), initialization of hidden states (Calixto and Liu, 2017), or by attention mechanisms (Libovický and Helcl, 2017; Calixto et al., 2016, 2017; Yao and Wan, 2020) and latent variables (Calixto et al., 2019).

Research has shown that the extra modality helps translation, especially when the input is incomplete (Caglayan et al., 2019, 2020; Imankulova et al., 2020) or ambiguous (Ive et al., 2019; Wu et al., 2019b). Wu et al. (2019a) hint that multimodal information could assist NMT in handling natural noise stemming from the speech recognition system, which is employed as the first step in their pipeline for speech-to-text translation from videos. Their results, however, were inconclusive, leaving the potential impact of multimodality on mitigating such noise unclear. Salesky et al. (2021) investigate the robustness of open-vocabulary translation by representing texts as images and applying optical character recognition to cover cases of noise such as misspellings. This is an interesting but orthogonal area of this work since no external visual information is used.

Therefore, it remains an open question whether MMT can perform better than NMT when dealing with noisy texts, and whether multimodality can be complementary rather than redun-

dant to previous text-based robustness techniques. The work of Caglayan et al. (2019) is the most closely aligned with our approach; however, they focused primarily on identifying when visual information is helpful. As such, they only performed experiments that compared NMT and MMT in the presence of unknown words consisting of placeholders used to mask out words in the source sentence. In contrast, we focus on multimodal models for realistic noise that includes in- and out-of-vocabulary words, such as misspellings or correctly spelled words used in an incorrect context.

3.3 Methods

In this section, we introduce our methods to improve and evaluate the robustness of the NMT and MMT models. Section 3.3.1 illustrates the multimodal cross-attention method used in the MMT model. In Section 3.3.2, we describe three techniques to inject realistic noise into training and test data. In Section 3.3.3, we introduce our error correction training method.

3.3.1 Multimodal Cross-attention

We use Transformer model (Vaswani et al., 2017) as our underlying MT model for both NMT and MMT. As described in Section 2.1.1, the attention mechanism works as a weighted sum of all value vectors V , with the weights dynamically computed based on the query vectors Q and key vectors K . The encoder employs self-attention layers where the query, key, and value come from the source hidden representation H_s . In the decoder, an extra cross-attention layer is appended, and the query comes from the target representation H_t while the key and value come from the encoder output (hidden output from the last source layer).

Transferring to the multimodal setting, visual features can be introduced in the self-attention or cross-attention module by replacing the text-only source representation H_s with a multimodal representation. Yao and Wan (2020) introduce multimodal self-attention by concatenating the

textual representation H_s and the visual features H_v to construct the query.

$$\text{Attention}_{\text{MMself}} = \text{Attention}([H_s; H_v]W^Q, H_sW^K, H_sW^V) \quad (3.1)$$

However, Caglayan et al. (2020) show that integrating visual features in the decoder results in better performance in the task of simultaneous multimodal translation. Therefore, in this work, we adopt a multimodal cross-attention method in a serial manner. In the decoder layer, a standard cross-attention is performed first on the representation of the source text, resulting in an intermediate textual context C_{txt} . Then, another cross-attention on the visual features is followed, where the query vector is from the textual context C_{txt} and the key and value vectors are from the visual features H_v .

$$\begin{aligned} C_{txt} &= \text{Attention}(H_tW^Q, H_sW^K, H_sW^V) \\ \text{Attention}_{\text{MMcross}} &= \text{Attention}(C_{txt}W^Q, H_vW^K, H_vW^V) \end{aligned} \quad (3.2)$$

3.3.2 Noise Injection

In previous work on noise injection, the perturbations are often arbitrary, which would result in unrealistic noise. To simulate natural noise in real-world applications, we add constraints to the random perturbations. We select three constrained noise injection methods that can be applied to both training and test data, with each method simulating one type of human-generated error:

Edit distance A word is randomly replaced by another word in the vocabulary where the edit distance between the two words is less than two characters. The edit-distance noise simulates the occurrence of confusable spellings (e.g., sat vs. seat) and also some grammatical errors (e.g., horse vs horses).

Homophones A word is randomly replaced by another word that shares the same pronunciation. We use the CMU Pronouncing Dictionary¹ to transform words into phonemes and find noisy substitutes with the same pronunciation. This simulates errors made by applications such as automatic speech recognition or by non-native speakers.

Keyboard (Belinkov and Bisk, 2018) A character in a word is randomly replaced with an adjacent key on the standard QWERTY keyboard. The keyboard noise simulates real-life typos when users accidentally press wrong keys while typing.

Table 3.1 shows an example of the three types of noise we experimented with. The edit distance and homophone noise types are applied at the word level, while the keyboard noise is on the character level. Word-level noise is more likely to break the sentence context even though the noisy substitutes are correctly spelled words. In contrast, character-level noise is likely to introduce misspelled words and increase the out-of-vocabulary (OOV) rate.

clean	a pink flower is starting to bloom .
edit-distance	a pink flower is staring to loom .
homophone	a pink flour is starting to bloom .
keyboard	a pink flower is starring to bloom .

Table 3.1: An example of noise injected. The noisy substitutes are marked in red.

In constructing the noisy training and test sets, we sample from three types of noise, following a uniform distribution, and apply only one type of noise to each sentence. To maintain the contextual integrity of the sentences, we avoid substituting function words that carry limited contextual information (e.g., articles and punctuation). Instead, we perturb words with more than two characters. The noise level is controlled by the hyperparameter $\mathbf{n}$, which defines the maximum number of words replaced with noisy counterparts per sentence. The noise injection procedure can be characterized as follows: given a source sentence $\mathbf{x} = [x_1, x_2, \dots, x_M]$ and a target translation $\mathbf{y} = [y_1, y_2, \dots, y_N]$, noise will be injected into the clean source sentence $\mathbf{x}$ to obtain its noisy variant $\mathbf{x}' = [x_1, \dots, x'_{a_i}, \dots, x_M]$, where a_i is the position of the noisy substitutes ($i = \{1, 2, \dots, n\}$).

¹<http://www.speech.cs.cmu.edu/cgi-bin/cmudict?in=C+M+U+Dictionary>

3.3.3 Error Correction Training

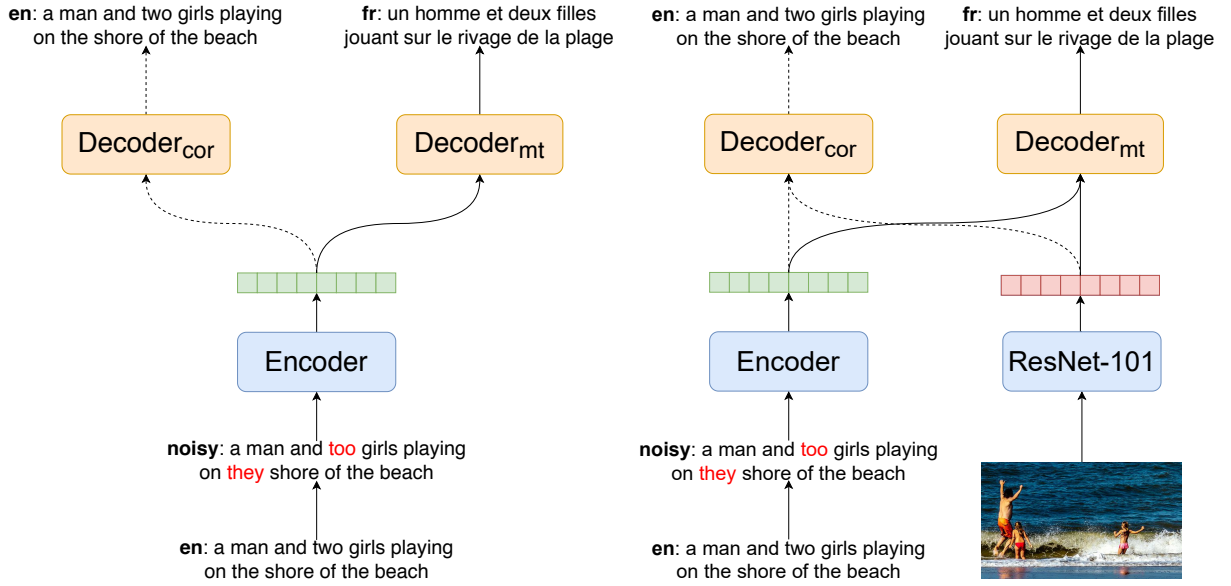


Figure 3.2: Illustration of the joint training of machine translation and error correction for NMT and MMT models. Left: NMT with error correction training. Right: MMT with error correction training.

To best leverage the noise-injected synthetic data, we introduce error correction training (Ng et al., 2014; Yuan and Briscoe, 2016) as a replacement for the conventional adversarial training method. The error correction training functions as an auxiliary task to enhance model robustness against noisy inputs. To achieve this, we incorporate a second decoder into the MT architecture, which is only used for the error correction task, and thus its parameters are frozen when optimizing the translation task. During training, the noisy sentence $\mathbf{x}'$ is encoded by the shared encoder, which is used in both translation and correction tasks, into the hidden state $\mathbf{h}'$. The hidden state $\mathbf{h}'$ is then passed to both decoders: the translation decoder generates the correct translation $\mathbf{y}$, while the correction decoder attempts to reconstruct the source sentence $\mathbf{x}$ by correcting the errors. This method is fully compatible with the MMT model, where the error correction decoder utilizes both visual and textual hidden states to recover clean source sentences. Figure 3.2 illustrates the architecture of the proposed method.

Compared to standard MT models, the error correction-enhanced models (which we refer to as *NMT-cor* and *MMT-cor* hereinafter) optimize two objectives: maximizing the probability of generating correct translations $P(\mathbf{y}|\mathbf{x}'; \boldsymbol{\theta}_{mt})$ and the probability of recovering clean source

sentences $P(\mathbf{x}|\mathbf{x}'; \boldsymbol{\theta}_{cor})$.

$$\begin{aligned} P(\mathbf{y}|\mathbf{x}'; \boldsymbol{\theta}_{mt}) &= \prod_{t=1}^N P(y_t|\mathbf{y}_{1:t-1}, \mathbf{x}'; \boldsymbol{\theta}_{mt}) \\ P(\mathbf{x}|\mathbf{x}'; \boldsymbol{\theta}_{cor}) &= \prod_{t=1}^M P(x_t|\mathbf{x}_{1:t-1}, \mathbf{x}'; \boldsymbol{\theta}_{cor}) \end{aligned} \quad (3.3)$$

The $\boldsymbol{\theta}_{mt}$ represents the parameters for the translation component (i.e., shared encoder and translation decoder) and the $\boldsymbol{\theta}_{cor}$ represents parameters for the error correction component (i.e. shared encoder and error correction decoder), with $\boldsymbol{\theta}_{mt} = \{\boldsymbol{\theta}_{enc}, \boldsymbol{\theta}_{mt_dec}\}$, and $\boldsymbol{\theta}_{cor} = \{\boldsymbol{\theta}_{enc}, \boldsymbol{\theta}_{cor_dec}\}$. We hypothesize that the auxiliary task of error correction could help the encoder with a noise-invariant representation, which would indirectly improve the translation of noisy sentences. During training, we jointly optimize the sum of translation loss and error correction loss, as shown in Equation 3.4:

$$\begin{aligned} \mathcal{L}_{mt}(\boldsymbol{\theta}_{mt}) &= \frac{1}{|\mathbf{D}|} \sum_{(\mathbf{x}', \mathbf{y}) \in \mathbf{D}} -\log P(\mathbf{y}|\mathbf{x}'; \boldsymbol{\theta}_{mt}) \\ \mathcal{L}_{cor}(\boldsymbol{\theta}_{cor}) &= \frac{1}{|\mathbf{D}|} \sum_{(\mathbf{x}', \mathbf{x}) \in \mathbf{D}} -\log P(\mathbf{x}|\mathbf{x}'; \boldsymbol{\theta}_{cor}) \\ \mathcal{L}(\boldsymbol{\theta}) &= \mathcal{L}_{mt}(\boldsymbol{\theta}_{mt}) + \lambda \mathcal{L}_{cor}(\boldsymbol{\theta}_{cor}) \end{aligned} \quad (3.4)$$

where $\lambda \geq 0$ is the factor that controls the weight of the error correction loss, and $\mathbf{D}$ represents the noise-injected data consisting of triples in the form of $(\mathbf{x}, \mathbf{x}', \mathbf{y})$.

3.4 Experiments

3.4.1 Datasets

For the experiment, we use the Multi30K dataset (Elliott et al., 2016), focusing on the En-Fr and En-De language pairs. This dataset is the standard benchmark for MMT and has been used in all open challenges on this topic (Specia et al., 2016b; Elliott et al., 2017a; Barrault et al., 2018). Following Caglayan et al. (2019), we use both the *train* and *valid* splits as our

training set. The *test2016-flickr* set is used as our development set for checkpoint selection and hyperparameter tuning. For evaluation, we test the models on both the *test2017-flickr* and *test2017-mscoco* sets (Elliott et al., 2017b). We employ a word-level vocabulary and build vocabularies for the source and target languages, as well as the vocabulary on noisy source texts. Therefore, there is no OOV word in the noisy training data, but the test data might still contain OOV words – noisy or not – with respect to the training. We use preprocessed data in Multi30K, which is lowercased, normalized, and tokenized with the *moses* toolkit (Koehn et al., 2007). In addition, we also perform experiments using a subword-level vocabulary (BPE), which led to further improvements.

On visual features extraction, we use the “bottom-up-top-down” (BUTD) features (Anderson et al., 2018a) extracted from a pre-trained Faster R-CNN ResNet-101 object detector. Each image is represented as 36 pooled feature vectors $V \in \mathbb{R}^{36 \times 2048}$, with each vector representing a local object region.

3.4.2 Models

NMT and MMT Models Our baseline NMT model is the standard Transformer model (Vaswani et al., 2017), with 6 layers for both the encoder and the decoder. Hyperparameters are tuned for the best performance on the development set. Specifically, the hidden state size is 512, and the feed-forward dimension is 1024. The number of attention heads is set to 4. A dropout rate of 0.3 is applied to both self-/cross-attention and the position-wise feed-forward layer. We also employ pre-layer norm (Nguyen and Salazar, 2019) to boost the convergence speed. The baseline MMT model follows the same architecture and hyperparameters as the baseline NMT model, except for the multimodal components. We use the serial multimodal cross-attention (Libovický et al., 2018), where an extra cross-attention sublayer is added in the decoder layer to perform attention over the visual features. We also experiment with GRU models (Cho et al., 2014b), following the hyperparameter settings of Caglayan et al. (2019).

Error Correction Models The error correction-enhanced NMT/MMT models adopt the same encoder and decoder as the baseline NMT/MMT models, except for a second decoder dedicated to the error correction task. During training, we compute the cross-entropy loss for both translation and error correction tasks, optimizing their loss jointly. In these models, the two losses are summed and optimized over the same batch. Through hyperparameter tuning, we determine the best λ value ($\lambda \in 0.2, 0.4, 0.8$) for various noise levels (number of noisy words $n \in 1, 2, 4, 6, 10$).

Training and Evaluation We use the ADAM optimizer (Kingma and Ba, 2015a) during training and adopt the *noam* learning rate scheduler (Vaswani et al., 2017) with a warm-up of 8000 steps. The training batch size is 64. The models are evaluated using the METEOR score (Denkowski and Lavie, 2014), which is the main metric for multimodal machine translation (Barrault et al., 2018). For the evaluation of error correction, we use ERRANT (Bryant et al., 2017) to compute the $F_{0.5}$ score. During evaluation, we select the checkpoint with the best performance on the development set and generate translation and correction using a beam search of size 12. All models are implemented using `nmtpytorch`² and `pysimt`³. Each model is run with three random seeds and the average results are reported. Each run takes approximately 2 hours to train on an RTX 2080 Ti GPU.

3.5 Results

3.5.1 Models Trained on Clean Data

We first evaluate the robustness to noise of standard NMT and MMT models that are trained without noise injection and error correction training applied. The models are **trained on clean data** and **evaluated on the noise-injected test data**. This setting represents regular models that are not specifically adapted to noise. Figure 3.3 presents the difference in METEOR

²<https://github.com/lium-lst/nmtpytorch>

³<https://github.com/ImperialNLP/pysimt>

(Δ METEOR) between standard MMT and NMT models tested on data with different noise levels. The Δ METEOR is consistently above 0 for both test sets in the two language pairs. As the noise level increases, the difference between the NMT and MMT models is larger, showing that the visual information in the MMT model leads to predictions that are more robust to noise.

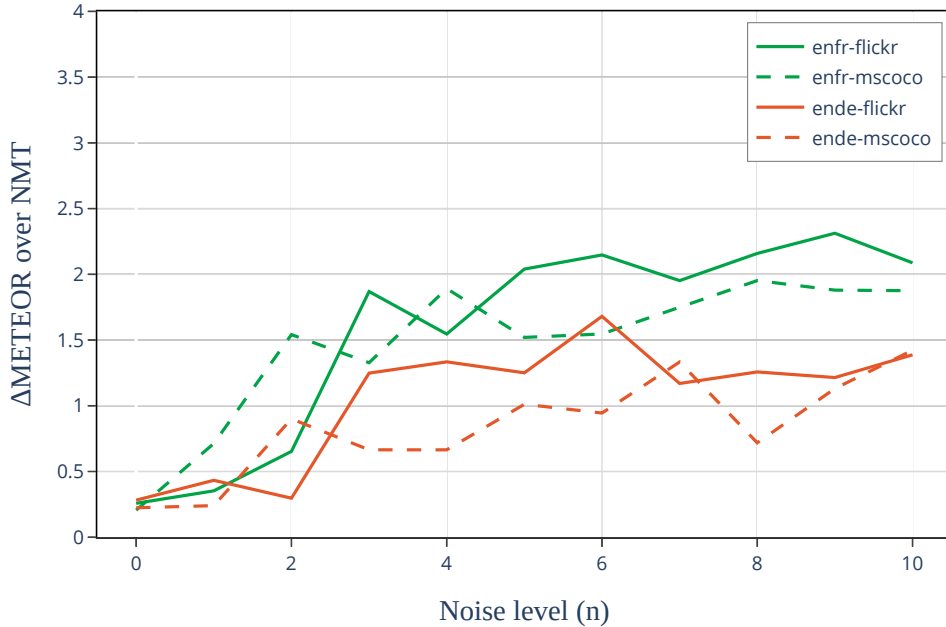


Figure 3.3: Performance gain from multimodality on different test sets when models are trained on clean data but tested on noisy data (Δ METEOR = MMT - NMT).

However, despite that the MMT model is more robust to noise at test time than the NMT model, its performance on noisy data is significantly degraded compared to that on clean data. Table 3.2 shows the METEOR scores of NMT/MMT models tested on different levels of noise. A general downward trend can be observed across both models on all test sets, which is further exacerbated w.r.t. the increase of noise level. This huge performance drop hints at the necessity of training models on noisy data for better robustness.

3.5.2 Models Trained on Noisy Data

To test models for their ability to adapt to noisy data, we **train models on data with added noise**, sampling the three types of noise in Section 3.3.2 and **test them on noisy test data**,

	flickr2017						mscoco2017					
	clean	$n=1$	$n=2$	$n=4$	$n=6$	$n=10$	clean	$n=1$	$n=2$	$n=4$	$n=6$	$n=10$
<i>en-fr</i>												
NMT	70.5	59.4	49.4	34.1	25.5	19.8	64.2	55.0	45.8	31.4	23.4	17.9
MMT	70.8	59.8	50.0	35.6	27.6	21.9	64.4	55.7	47.3	33.3	25.2	19.8
<i>en-de</i>												
NMT	52.3	44.5	37.0	25.8	19.6	16.0	47.5	41.2	34.3	24.6	19.2	15.1
MMT	52.6	45.0	37.3	27.1	21.3	17.4	47.7	41.4	35.2	25.3	20.2	16.5

Table 3.2: Results in METEOR scores of standard NMT/MMT models tested on different levels of noise.

with noise added in the same fashion.

3.5.2.1 Transformer models

	flickr2017					mscoco2017				
	$n=1$	$n=2$	$n=4$	$n=6$	$n=10$	$n=1$	$n=2$	$n=4$	$n=6$	$n=10$
<i>en-fr</i>										
NMT	64.2	60.2	55.2	51.8	49.4	58.3	54.3	48.8	45.7	43.2
MMT	64.7	61.0	56.8	53.7	51.1	59.3	55.4	50.1	47.8	45.2
NMT-cor	64.9	61.6	57.4	54.7	55.0	59.2	55.2	51.4	48.0	47.2
MMT-cor	65.2	62.2	59.0	56.7	55.5	59.6	56.4	52.4	50.0	48.9
<i>en-de</i>										
NMT	47.2	44.3	40.2	38.4	36.7	43.5	40.2	36.8	34.0	32.5
MMT	47.7	45.2	41.3	39.3	37.6	43.9	41.0	37.9	35.1	33.9
NMT-cor	47.9	45.6	42.9	41.4	41.1	44.2	41.9	38.4	36.8	36.2
MMT-cor	48.0	46.1	43.5	42.5	41.8	43.9	42.3	39.7	38.2	37.4

Table 3.3: Results in METEOR scores of models trained and tested on different levels of noise.

Performance on noisy test data The results in the METEOR score are shown in Table 3.3. The training on noisy data is equivalent to the “adversarial training” experiments explored in previous studies (Belinkov and Bisk, 2018; Karpukhin et al., 2019). In this setting, a text-only NMT model still suffers from significant performance degradation as the number of noisy words grows, for example, dropping from 70.5 METEOR on clean test data to 49.4 under the noisiest setting for en-fr translation on the flickr2017 test set. A drop is also observed for the MMT model, however, it is smaller for both language pairs and test sets. As n becomes larger,

the gain from the visual context is more obvious, showing that additional context in the form of image features is increasingly important for translation when the quality of the textual input is degraded. Comparing the same results in Table 3.2, we notice that training on noisy data improves the performance on noise to a significant degree, which confirms the effectiveness of adversarial training in improving robustness.

With the addition of error correction training, both the NMT-cor and MMT-cor models further improve their performance, with NMT-cor even outperforming the base MMT model. The MMT-cor model performs better than both the NMT-cor and base MMT models, demonstrating that the improvements from error correction and visual cues are complementary. Similar to the benefit from visual features, the difference between models with and without error correction training becomes larger as the noise level increases.

Performance on clean test data In addition to the performance on noisy texts, another important aspect when measuring robustness is to evaluate whether the performance of the models on clean data is harmed when the model is adapted to the noisy data. Table 3.4 shows the performance drop on the clean Flickr2017 En-Fr test set when models are trained with different levels of noise, compared to the baseline NMT model trained with clean data only. All models trained on fully noisy data show a significant performance drop on the clean test set, despite the fact that the introduction of visual features and error correction could mitigate such a drop. In addition, the trend is the same for the models: When a larger proportion of noise is added to the training data, the model suffers from a greater performance drop on the clean test set.

To balance the improvement in robustness to noise with performance on clean texts, we follow Karpukhin et al. (2019) and **train models on a mixture of noisy and clean data (0.5/0.5)**. Compared to training on noisy data, mixed training improves the performance on clean data, with the largest drop in METEOR being only 2.4. The smaller drop shows that mixing clean and noisy training data can help improve the model’s robustness while maintaining the performance on clean data. Both MMT and MMT-cor show a similar performance drop to the base NMT

$n =$	1	2	4	6	10
NMT	↓1.5	↓2.4	↓5.2	↓9.3	↓19.8
MMT	↓0.7	↓1.9	↓4.5	↓8.6	↓18.1
MMT-cor	↓0.8	↓1.7	↓4.4	↓7.7	↓15.5
NMT-mix	↓0.2	↓1.0	↓1.4	↓2.0	↓2.3
MMT-mix	↓0.2	↓0.7	↓1.7	↓2.1	↓2.4
MMT-cor-mix	↓0.0	↓0.4	↓0.9	↓1.7	↓2.1

Table 3.4: Performance drop on clean test set (the lower the better) when models are trained on mixed data, compared to standard models trained on clean data.

model, which indicates that the use of visual context and error correction training does not harm performance on clean texts.

3.5.2.2 GRU models

In Table 3.5, we present the results for the GRU models trained and tested on noisy data. Similarly to transformer models, GRU models also benefit from multimodality and error correction training, and the improvement is larger on noisier data.

	flickr2017					mscoco2017				
	$n=1$	$n=2$	$n=4$	$n=6$	$n=10$	$n=1$	$n=2$	$n=4$	$n=6$	$n=10$
<i>en-fr</i>										
NMT	64.9	61.1	55.9	53.0	50.7	59.5	55.4	49.2	45.8	43.7
MMT	65.3	61.9	57.5	54.5	52.0	59.7	56.5	50.9	47.4	45.2
NMT-cor	65.2	61.8	57.3	54.6	53.1	59.8	55.9	50.6	48.0	45.8
MMT-cor	65.4	62.5	58.4	56.0	54.3	60.3	56.6	51.5	49.0	47.3
<i>en-de</i>										
NMT	48.0	45.3	41.5	39.7	36.8	43.5	40.9	36.2	34.6	30.7
MMT	48.5	45.9	42.5	40.6	39.4	43.9	41.3	37.7	35.3	33.7
NMT-cor	48.6	46.3	43.1	40.7	39.1	44.1	41.7	37.4	35.5	33.3
MMT-cor	48.5	46.7	44.0	42.6	41.3	44.3	42.0	39.0	37.3	35.6

Table 3.5: Results for GRU models trained and tested on different levels of noisy data.

In Table 3.6, the performance drop for GRU models on clean data is presented. Both MMT and MMT-cor show a lower drop than the NMT baseline, confirming that the improved robustness on noisy data does not come at the cost of the ability to translate clean data.

These results with GRU models further confirm that both multimodality and error correction

$n =$	1	2	4	6	10
NMT-mix	↓0.4	↓0.5	↓1.5	↓2.3	↓3.1
MMT-mix	↓0.2	↓0.9	↓1.3	↓2.2	↓2.4
MMT-cor-mix	↓0.2	↓0.6	↓1.6	↓1.9	↓2.7

Table 3.6: Performance drop (the lower the better) on the clean Flickr2017 En-Fr test set when GRU models are *trained on mixed data* but *tested on clean data*.

training improve translation robustness and can be generalized to different models.

3.6 Analysis

3.6.1 Robustness on Unseen Noise

Since in real-world applications the noise distribution at test time is unknown, we evaluate models using different noise proportions and types at training and test time. For the former, we test the same model trained with mixed data ($n=4$) on various test sets created with different values of n . For the latter, we test the same model ($n=4$) on the test set where words are randomly replaced with unknown tokens (i.e., “[UNK]”) to simulate unseen noise (e.g., noisy words from different corpora or domains such as new emojis). Table 3.7 shows the results for both cases.

$n =$	1	2	6	10
NMT	62.5	59.3	51.6	49.2
MMT	62.9	60.1	52.8	51.0
MMT-cor	64.1	62.0	55.5	53.8
UNK=	1	2	3	4
NMT	55.5	46.7	38.7	31.6
MMT	57.0	48.8	41.2	34.8
MMT-cor	57.9	49.9	42.6	36.1

Table 3.7: Performance of models trained on noisy data with $n=4$ but tested on data with different noise proportions and noise types. All models are tested on Flickr2017 En-Fr.

The overall trend is similar to the case where the train/test noise is the same: models with visual information and error correction training achieve better performance. The METEOR

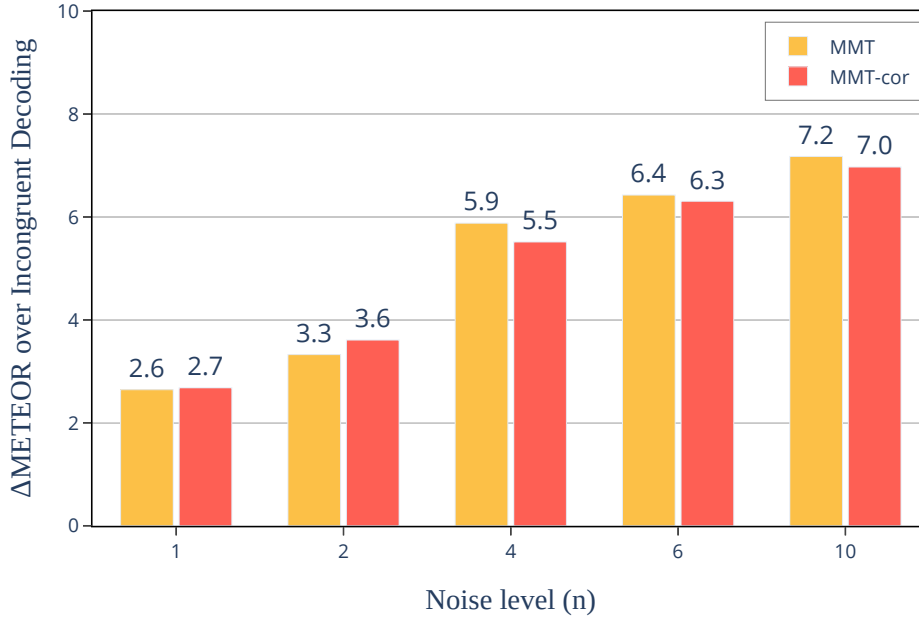


Figure 3.4: Performance gap in METEOR score between congruent decoding and incongruent decoding ($\Delta\text{METEOR} = \text{congruent} - \text{incongruent}$).

score for the train / test noise proportion mismatch is close to the score in Table 3.3 under the same noise proportion, showing that the models are robust to unknown noise distributions. In terms of evaluation of unknown noise types, the MMT model outperforms the NMT model, indicating the better ability of the MMT model to handle unseen noise.

3.6.2 Visual Sensitivity

To further probe the effect of visual information on the MMT and MMT-cor models, we apply the *incongruent decoding* evaluation approach (Elliott, 2018; Caglayan et al., 2019) by feeding multimodal models with incorrect visual features at test time, i.e., features taken from a different test sample. The expectation is that the multimodal model will suffer due to the incorrect visual context, performing worse compared to using the correct visual features. The drop in score after applying incongruent decoding can indicate how much visual features contribute to the performance when using the correct visual features. Figure 3.4 shows the performance gap between congruent decoding and incongruent decoding.

The ΔMETEOR is always positive for both the MMT and MMT-cor models, and this difference

is amplified by a larger noise ratio in the test data, reaching up to 7.2 METEOR scores when $n=10$. We note that the Δ METEOR for the MMT-cor model is similar to the MMT model, but slightly lower, indicating that error correction training helps the model recover from incorrect image features to a very small extent on noisier data.

3.6.3 Effect of λ

In the proposed method, λ is an important hyperparameter that controls the weight of the error correction training for NMT-cor and MMT-cor models. Therefore, we show the performance of translation and error correction tasks for different values of λ in Figure 3.5.

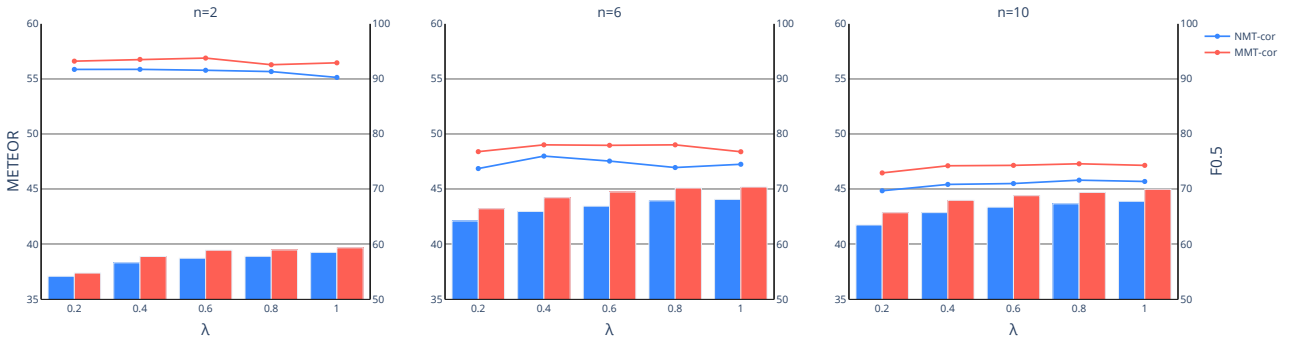


Figure 3.5: Effect of λ on translation and error correction tasks. Lines: translation performance in METEOR score. Bars: error correction performance in $F_{0.5}$ score. The results are tested on MSCOCO2017 En-Fr data.

In terms of translation, the performance for both the NMT-cor and MMT-cor models follows the same trend: the METEOR score first increases and then drops as λ increases. This is reasonable since error correction is an auxiliary task, and a large weight for the error correction task might harm models' ability to translate well. Nevertheless, the optimal λ value is different for different levels of noise. Higher values of λ help translating noisier texts. Regarding error correction, the increase of λ always leads to better performance.

3.6.4 Error Correction Quality

To understand whether visual information can also benefit error correction, we compute the span-based correction $F_{0.5}$ score, which is commonly used as an evaluation metric in the Grammatical Error Correction task (Dahlmeier and Ng, 2012). The $\langle \text{noisy}, \text{corrected} \rangle$ and $\langle \text{noisy}, \text{clean} \rangle$ pairs are first transformed into two lists of edits, where adding/replacing/deleting a word at any position counts as one edit. The evaluation is then performed by calculating the precision/recall/ $F_{0.5}$ between these edit sets.

	flickr2017			mscoco2017		
	Prec	Rec	$F_{0.5}$	Prec	Rec	$F_{0.5}$
$n=1$						
NMT-cor	41.9	52.5	43.7	45.1	51.4	46.2
MMT-cor	43.3	54.0	45.1	46.5	53.8	47.8
$n=2$						
NMT-cor	56.7	62.2	57.7	53.2	56.2	53.8
MMT-cor	57.0	63.3	58.1	52.9	56.4	53.6
$n=4$						
NMT-cor	66.6	69.1	67.1	65.7	66.7	65.9
MMT-cor	67.7	71.5	68.5	66.1	67.6	66.4
$n=6$						
NMT-cor	68.7	70.0	69.0	67.0	66.1	66.8
MMT-cor	70.4	71.8	70.7	68.3	67.6	68.2
$n=10$						
NMT-cor	72.5	73.2	72.6	67.1	66.4	67.0
MMT-cor	73.9	74.5	74.1	69.8	68.6	69.6

Table 3.8: Error correction score in $F_{0.5}$ for both NMT-cor and MMT-cor models.

We report the results in Table 3.8 for both NMT-cor and MMT-cor models trained on different values of n . The MMT-cor model outperforms the NMT-cor model, with an improvement of up to +1.7 and +2.6 $F_{0.5}$ on the two test sets. This improvement indicates that visual features can also be beneficial for error correction performance, showing a potential for the task of multimodal error correction, which has yet to be explored.

3.7 Qualitative Examples

We provide two qualitative examples of visual features and error correction training that help the model handle input noise in Table 3.9. In the first example, the source sentence is injected with the “edit-distance” noise, with “are” and “orange” replaced with “art” and “strange” respectively. Both NMT and NMT-cor models fail to include “orange” in the translation, as it is difficult to recover from this error without visual information, while the MMT-cor model can generate the correct output. The source sentence in the second example is injected with the “keyboard” noise, with “man” replaced with “kan” and “bicycle” replaced with “bicyclcd”. Although the training data is injected with the same types of noise, the NMT model fails to translate correctly. The reason might be that “bicycle” has multiple noisy variants, such as “bicyclcd”, “bocycle”, etc., so the NMT model can hardly learn a strong relationship between “bicyclcd” and “vélo” (translation of “bicycle”). However, the NMT-cor model could relate “bicyclcd” with “bicycle”, which helps predict the correct translation “vélo”.

	SRC: women are playing lacrosse with an orange ball . NSY: women [art] playing lacrosse with an [strange] ball . NMT: des femmes jouent au lacrosse avec une balle <u>étrange</u> . NMT_{cor}: des femmes jouent au lacrosse avec une boule <u>étrange</u> . (<i>women are playing lacrosse with a strange ball.</i>) MMT_{cor}: des femmes jouent à la lacrosse avec une balle orange . REF: des femmes jouent à la crosse avec une balle orange . (<i>women are playing lacrosse with an orange ball .</i>) COR-NMT: women are playing lacrosse with an <u>old</u> ball . COR-MMT: women are playing lacrosse with an orange ball .
	SRC: a man with his bicycle selling his products on a street NSY: a [kan] with his [bicyclcd] selling his products on a street NMT: un homme avec son <u>casque</u> vendant ses produits dans une rue (<i>a man with his helmet selling his products on a street</i>) NMT_{cor}: un homme avec son vélo vendant ses produits dans une rue MMT_{cor}: un homme avec son vélo vendant ses produits dans une rue REF: un homme avec son vélo vendant ses produits dans une rue (<i>a man with his bicycle selling his products on a street</i>) COR-NMT: a man with his bicycle selling his products on a street COR-MMT: a man with his bicycle selling his products on a street

Table 3.9: Qualitative examples for both translation and error correction, where noise is indicated by the words in square brackets. Underlined and bold words highlight the bad and good lexical choices, respectively. NSY: noisy sentence. COR-*: corrected sentence (output from the error correction decoder).

In Figure 3.6, we present the attention map of the MMT-cor system when generating the translation. The input is injected with noise by substituting “sit” with “sheet”, and “wine”

with “wire”. When generating “sont assises” (are sitting), although the attention on the input text still mainly focuses on the noisy word “sheet” (with a small proportion focusing on the preposition “at”), the visual attention can focus on the people in the image; therefore, the model obtains the correct information from the visual input and can generate the correct translation. Similarly, the model generates “vin” (wine) by attending to the glasses in the images and is not distracted by the noisy input word “wire”.

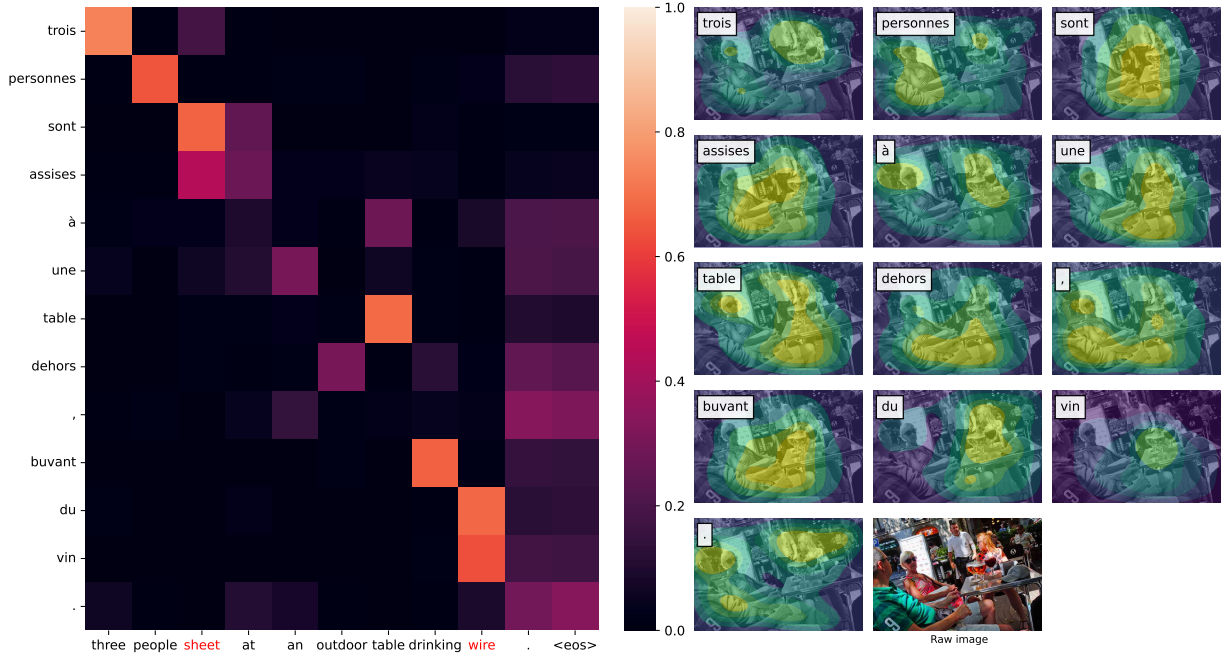


Figure 3.6: Attention map of the MMT-cor system on input texts and visual features when generating the translation from noisy input with the target decoder.

Figure 3.7 shows the attention map in the same example when generating the correction of the noisy input. When correcting the misspelled word “sheet”, the error correction decoder instead of focusing on the input word, it shows more attention to the surrounding words “at” and “an”, which helps generate the correct output “sitting”.

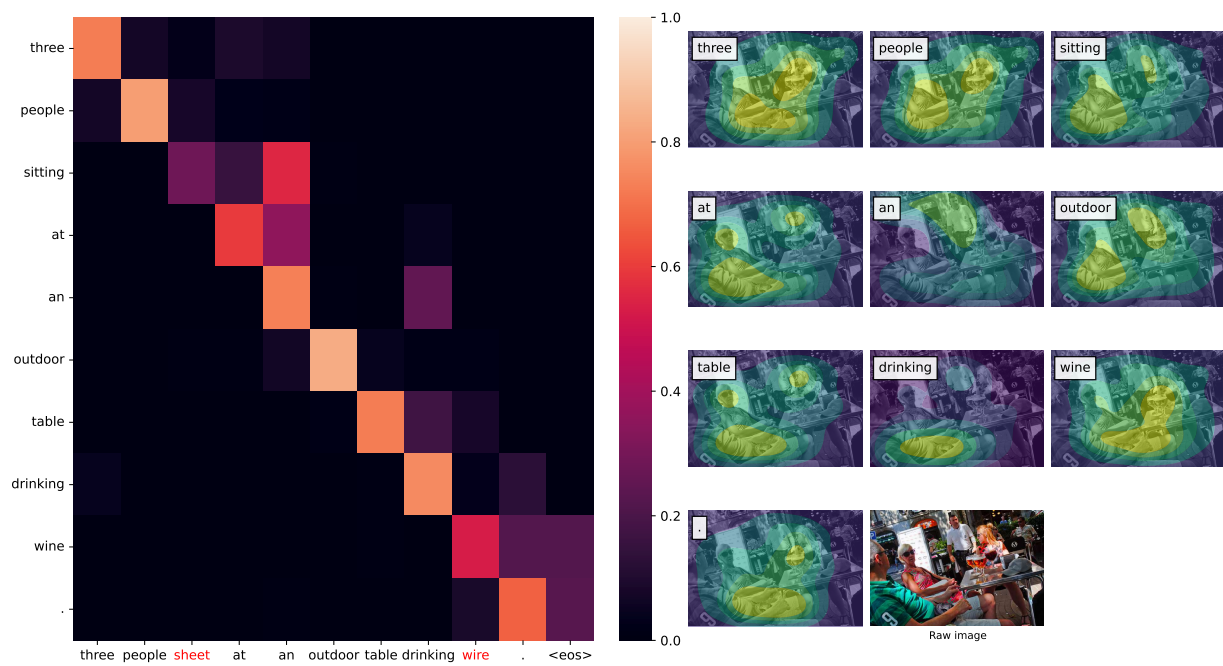


Figure 3.7: Attention map of the MMT-cor system on input texts and visual features when generating the error correction from noisy input with the correction decoder.

3.8 Conclusions

In this chapter, we propose exploring visual cues in order to improve model robustness to noise in machine translation. We combine adversarial training on artificially generated noisy examples with visually informed multimodal machine translation. By training multimodal models on noisy data, we show that **extra visual context can improve the robustness of translation in both known and unseen noise**. We also propose a novel error correction training method, jointly optimizing the translation model with an auxiliary objective to correct input errors, which we show can further improve the robustness of both text-only and multimodal translation models, proving that **the effects of visual context and textual context are complementary rather than redundant**.

Revisiting the research questions, this chapter primarily focuses on exploring how an extra modality can help tackle textual noise. In our experimentation, we show that the multimodal machine translation model consistently outperforms the uni-modal model, regardless of the training setting, standard training, or adversarial training. Furthermore, the success of error correction training also acts as a demonstration to show the potential of applying perturbation removal from text to improve robustness, supporting the method in Chapter 5, which removes latent noise from the hidden state of the model to boost adversarial robustness. In the next chapter, we will further incorporate conversational context in addition to visual information to handle textual noise, which evaluates on both naturally noisy and synthetically perturbed data.

Chapter 4

Better Noise Robustness with Multimodal Conversation Modeling

In Chapter 3, we explored methods to improve model robustness to noise by leveraging images and error-corrected texts on three types of injected noise. However, in real-world applications, user-generated content often contains a more diverse range of noise types. In this chapter, we focus on real-world noisy texts on social platforms where a more diverse noise distribution is present in the texts. Furthermore, we look into additional contextual information, conversation history, which can provide useful cues for the model to interpret and mitigate noisy texts. In this chapter, we focus on a critical problem on social platforms — topic derailment — where conversations deviate from their original subjects, and we explore how visual information and conversational context can help address the issues caused by textual noise.

Conversations on social media frequently go off-topic, evolving into unrelated or sometimes toxic exchanges. Detecting such cases of topic derailment is crucial to ensure healthy interactions online. However, comments on social platforms can contain various forms of textual noise, such as orthographic variants, obfuscations, leet speak, code-switching, etc, making it harder to model conversation topics and classify conversation derailment. While previous work has focused on analyzing textual dialogues that have devolved into toxic content, the scope of derailment types is much broader, extending beyond toxic content to spam, bot-generated

content, tangential comments, etc. In addition, existing work disregards conversations that involve visual information (i.e., images or videos), which are prevalent on most platforms.

In this chapter, we take a broader view of conversation derailment and propose a new challenge: detecting derailment based on the “change of conversation topic”, where the topic is defined by an initial post containing both text and an image. To address this challenge, we (i) create the first Multimodal Conversation Derailment (MCD) dataset, and (ii) introduce a new multimodal conversational architecture (MMConv) that utilizes visual and conversational contexts to classify comments for derailment detection. Experiments show that MMConv outperforms previous text-based approaches to detect conversation derailment, as well as general multimodal classifiers. Our analysis further validates that MMConv is also more robust to textual noise by leveraging richer multimodal and contextual information.

This chapter is based on the paper “Multimodal Conversation Modeling for Topic Derailment Detection” published in *Findings of EMNLP, 2022* (Li et al., 2022). The proposed dataset is available at <https://github.com/Nickeilf/Multimodal-Conversation-Derailment>.

4.1 Introduction

Online conversations on social media can easily go off-topic (Churchill and Bly, 2000; Shepherd et al., 2015) and result in divergent and even harmful exchanges, ranging from off-topic discussions to spam information and personal attacks. Work in this field has focused on detecting or forecasting conversations that derail into toxic content (Zhang et al., 2018a,b; Chang and Danescu-Niculescu-Mizil, 2019; Kementchedjhieva and Søgaard, 2021; Lambert et al., 2022). However, such work is limited in two ways: 1) derailment is not limited to toxic comments or personal attacks; it can include topic shifts, spam posts, among others; 2) conversations in real applications can include modalities other than text, such as image, audio, and video, and the multimodal context can be crucial in deciding whether a conversation has derailed.

In this work, we focus on a broader view of detecting conversation derailment based on the change of conversation topic, making it a more challenging task, and propose the task of

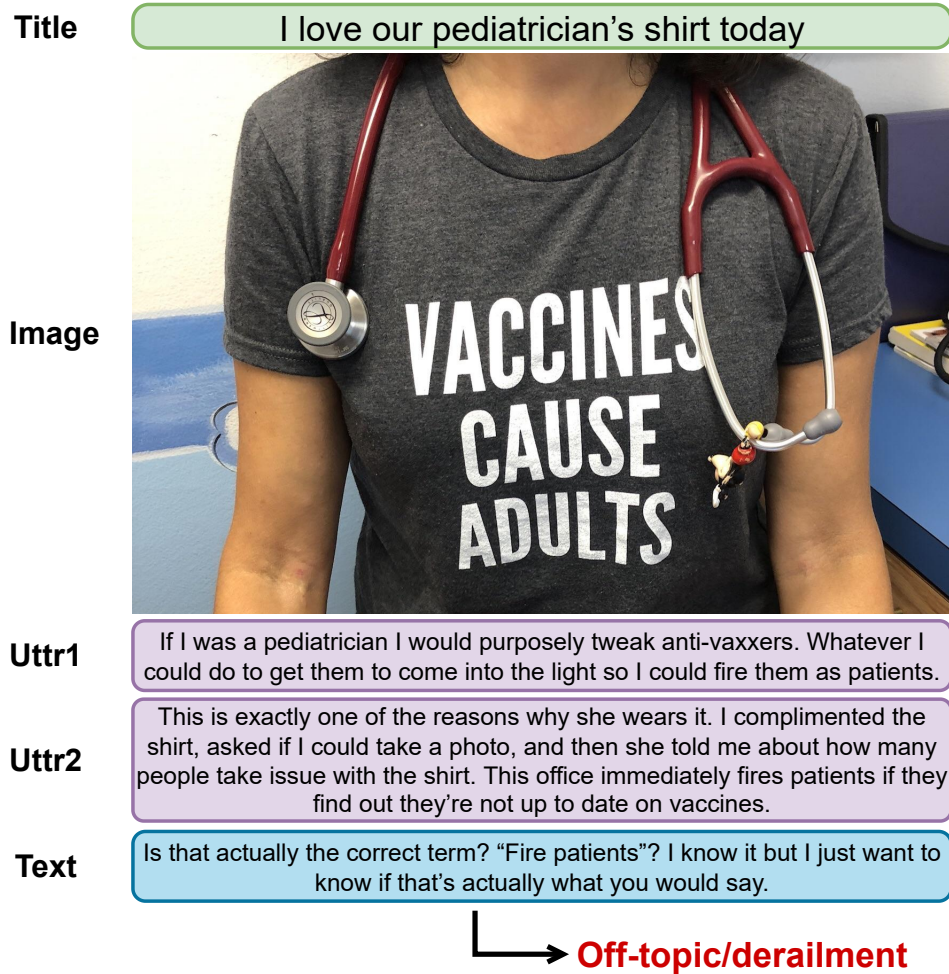


Figure 4.1: An example of multimodal conversation derailment detection. The conversation starts with a title and an image, with a sequence of utterances replying to each other and a target text which is classified to be a derailment or non-derailment.

multimodal derailment detection. Figure 4.1 shows an example of multimodal conversation derailment detection, which includes multiple input types. We collect data from Reddit and use crowd annotation to identify derailments, creating the first Multimodal Conversation Derailment (MCD) dataset. The MCD dataset contains various types of common noise, such as obfuscations, emojis, etc. To the best of our knowledge, this is the first work introducing multimodality in conversation derailment detection. In addition, we propose a new multimodal conversational (MMConv) architecture that utilizes image and conversational contexts via a hierarchical structure to improve the performance of derailment detection. Furthermore, we also perform a controlled noise injection and measure the robustness to noise under different levels of noisy settings. To summarize, our main contributions are:

- We propose a new task: multimodal conversation derailment detection, and create a new dataset for this challenging task.
- We propose a new hierarchical model architecture – MMConv – that uses visual and conversational contexts. Experiments show that our model outperforms existing multimodal and conversational architectures in this task.
- We evaluate the model’s robustness against noisy texts, showing that images and conversations can provide useful context and improve model robustness, regardless of the source of noise and quantity of noise in texts.

4.2 Related Work

4.2.1 Conversation Derailment Detection

Previous work in conversation derailment detection focuses on derailment due to antisocial behaviors. Zhang et al. (2018a) propose the Conversations Gone Awry dataset and introduce the task of forecasting antisocial behaviors in conversations. Chang and Danescu-Niculescu-Mizil (2019) propose the Reddit CMV dataset and a CRAFT model that can potentially provide early warnings regarding abusive content. Kementchedjhieva and Søgaard (2021) further introduce dynamic training into BERT-based derailment forecasting models. Other studies investigate general toxicity in conversations (Almerekhi et al., 2020; Cécillon et al., 2020; Ive et al., 2021; Vidgen et al., 2021; Lambert et al., 2022) regardless of conversation derailment. In this work, we study derailment detection from a broader angle, based on the change of conversation topics. The work by Park et al. (2021) is the closest to ours. They study conversations violating a set of community norms, such as uncivil behavior, off-topic content, etc. However, their work only considers textual conversations, whereas we look at multimodal conversations, which expands the possible application area.

4.2.2 Conversation Modeling

Existing approaches to modeling conversations can be divided mainly into two categories: hierarchical encoders and concatenation-based encoders. In hierarchical models, each utterance in the conversation is encoded first into an utterance vector, and a second context encoder aggregates these vectors to form a conversation representation. Sordoni et al. (2015) propose the HRED architecture, which uses two RNNs as the utterance encoder and context encoder. Serban et al. (2017) extend HRED with VHRED by injecting a latent variable into the context vector at each utterance to improve long-range coherence. Subsequent work has also explored different recurrent models for the utterance and context encoders, including GRUs (Yang et al., 2016), and LSTMs (Chang et al., 2019). Recent research also focuses on transformer-based hierarchical architecture, where a second self-attention layer is applied above the token-level transformer to capture inter-utterance dependencies (Zhang et al., 2019a,b; Pappagari et al., 2019; Santra et al., 2021), which is shown to outperform their RNN predecessors.

Concatenation-based models treat the entire conversation as a single sequence of tokens, separated by special tokens to differentiate utterances, and feed it to Transformer-based (Vaswani et al., 2017) pre-trained language models to generate a vector representation (Zhong et al., 2021). Lai et al. (2020) use a BERT encoder for conversation state tracking with utterances concatenated. Furthermore, auto-regressive decoder-only LMs, such as GPT, naturally consume the concatenated utterances and predict the next token. Although this approach can be more effective, it would require more computational resources to handle the long inputs, which could potentially be limited by the token size of the model (e.g., the max token length for BERT is 512). Therefore, research in this direction has focused on extending the context window with techniques like sliding-window attention (Beltagy et al., 2020), FlashAttention (Dao et al., 2022), enabling conversational models like DialoGPT (Zhang et al., 2020), DialogLM (Zhong et al., 2022), etc. (Shuster et al., 2022; Thoppilan et al., 2022), that can process long conversations with 8k to 32k tokens. In this work, we adopt the first type of approach to model the conversations hierarchically, due to resource efficiency constraints.

4.3 Multimodal Conversation Derailment Detection

4.3.1 Task Definition

On popular social platforms such as Reddit, a conversation often starts with an original post, which consists of a segment of text (title), sometimes with images (photo) or a paragraph of text (content). Users can then either comment under the post or reply to the comments. Therefore, the post forms a tree structure, where each branch can be seen as a conversation. Following this natural feature of a conversation, we define the multimodal dialogue derailment task, which has four types of inputs: a **title** T , an **image** I , a sequence of **previous utterances** (shortened to “utterances” in later paragraphs) $U = [U_1, U_2, \dots, U_n]$, and a **target text** X . The title and the image are posted by the same user and are complementary to convey the topic. Each utterance U_i is the reply to the previous utterance U_{i-1} , and the target text X is the reply to the last utterance U_n . The utterance sequence can be empty, and in this case, the target text is the direct reply to the title/image. This task aims to detect whether the target text has derailed the conversation. Specifically, the target text is considered a derailment if:

1. The target text does not directly relate to the initial topic. The initial topic can include the subject of the image, the image itself (its quality, color, lighting, etc.), and the title itself (the way it is formulated).
2. The target text refers to the content of the image or title, but starts a different conversation topic on them.
3. The target text is spam (e.g., advertising, unintelligible, etc.).
4. The target text is toxic (e.g., personal attack, insulting, etc.).

Figure 4.2 shows examples of the four types of derailments.

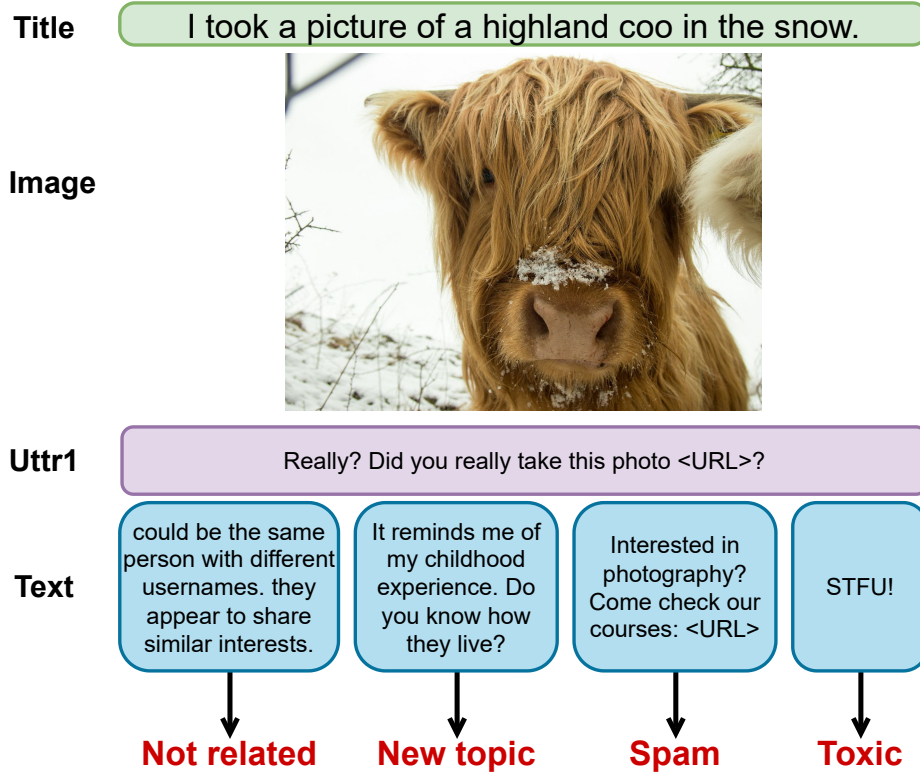


Figure 4.2: Example of different types of derailments.

4.3.2 Data Collection

Our data is sourced from Reddit using the Pushshift API¹ (Baumgartner et al., 2020). The data is taken from two subreddits with abundant images: /r/itookapicture and /r/pics. To select conversations with potential off-topic target texts, we focus on conversations based on three criteria:

1. Target text with keywords "off-topic", "off topic"².
2. Target text removed by subreddit moderators. We first find target texts with "[removed]" as the content using Reddit API³, and restore the original texts before moderation using Pushshift.
3. Target text with a Reddit score below -10. The score is calculated with the number of upvotes minus the number of downvotes.

¹<https://github.com/pushshift/api>

²We remove the sentences containing these keywords in the text to avoid potentially biasing the annotators.

³<https://github.com/praw-dev/praw>

For potential negative examples (on-topic target text), we select the target text with a score above 10. After collecting the target texts, we use the Reddit API to iteratively fetch previous utterances in the reply chain, until the original post containing the title and the image.

4.3.3 Annotation Procedure

The data is annotated by qualified US-based annotators from Amazon Mechanical Turk (AMT). We set up an annotator qualification process, and workers from the AMT platform must pass all of the following qualifications before they are allowed to annotate the data:

- The worker must be an adult since the data contains potential offensiveness.
- The worker’s number of approved annotations must be above 2,000 on the AMT platform.
- The worker’s approval rate must be above 95% on the AMT platform.
- The worker must pass a qualification test of 10 random conversations from the gold data and achieve an accuracy above 80%.

After passing all the above qualifications, the worker is granted access to be a qualified worker. Note that we also perform quality control during the annotation and block annotators with poor annotations. Once a worker is blocked, they can no longer annotate more data.

For each conversation, the annotator is asked to label whether the target text is a derailment with respect to the initial topic of the conversation, which is defined by the title and the image. The possible labels include “derailment”, “non-derailment”, and “skip (hard to decide)”. If a derailment is chosen, the annotator is asked to select the reason for the derailment. The possible derailment labels are: “unrelated to title”, “unrelated to image”, “unrelated to title and image”, “new conversation topic”, “spam comment”, and “toxic comment”.

Regarding quality control, we include gold data with pre-annotated labels in the annotation process. The annotators’ performance on this gold data is monitored during the annotation. If an annotator skips more than 10% of conversations or has an accuracy below 60% on the gold

Understanding Off-topic Online Conversation

The goal of this experiment is to understand how online conversation gets off-topic and derails into discussion of unrelated topics.

Device disclaimer: The platform is best viewed in computer browsers (mobile browsers might not load correctly)

Content Disclaimer: A small proportion of texts are toxic, and a small number of images might be NSFW. You can stop the annotation at any point.

Instructions

Each sample is an online conversation, which consists of a **title**, an **image**, a sequence of **history comments**, and a **target comment**. Your task is to determine **whether the target comment is unrelated to the initial topic of the post** (i.e. the topic that the post intended to convey). The initial topic of the conversation is determined by both the title and the image. The target comment is the one we want to identify whether it is off-topic.

The history comments are provided only for a better understanding of the conversation and should NOT define the initial topic. Read the target comment first. If that's enough to determine whether it is off-topic, you can skip reading the history comments. If not, use the history comments to help your decision.

Derailment identification

Derailment is always judged with respect to the content of the title and image. A target comment is considered as a derailment if:

1. The target comment does **not directly relate to** (e.g. without the need of extra knowledge, imagination, etc.) the **initial topic** of the title and image. The initial topic can include the subject of the image, the image itself (its quality, color, lighting, etc.), and the title itself (the way it is formulated). For example, the title/image is about Trump campaigning, but the target comment is about Trump's wife - who is not related to the initial topic. Then this target comment should be classified as a derailment.
2. The target comment talks about an object or person or event in the image/title, but **starts a different conversation topic** on them, like sharing a story related to the object/person/event. For example, the title/image is about LEGO price, but the target comment shares a childhood story of LEGO. In this case, it should be classified as a derailment because it introduces a different topic.
3. The target comment is a **spam** comment (e.g. advertising, unintelligible bot, etc.).
4. The target comment is a **toxic** comment (e.g. personal attack, insulting, etc.).

If you have any feedback regarding the project, you can fill this [Feedback Form](#).

Detailed Instructions

Title: How it started vs How it's going for the Irwin legacy



History 1: The brunette is way hotter!
 History 2: ...Okay, thanks for your observation?
 History 3: Tell me she's not tho...
 History 4: That's not really the point
 History 5: What is the point?
 History 6: Honestly, that nothing's wrong with noticing attractiveness, but that it's kinda unnecessary to point it out (to the insult of the woman on the left) when the post is about something else entirely. I noticed she was prettier too, but commenting on a woman's sexual appeal (especially comparing it to another woman's) when it's nowhere near related to the topic seems off to me.

Target Comment: Honestly, I don't know how you would determine the topic of a photo? What's front and center, probably.

1. Is the target comment a derailment with respect to the initial topic of the title/image?

The history does not need to match the initial topic because the focus is the target comment.

☒ Derailment (off-topic).
☐ Non-derailment (on-topic).
☐ Skip

2. On a scale of 0-5, how confident are you about your answer?

0 ————— 5

3. Why does the target comment derail?

☐ It is unrelated to the title
☐ It is unrelated to the image
☒ It is unrelated to neither title or image
☐ It is related to the title/image, but it might start a new topic
☐ It is a spam comment (e.g. advertising, bot, etc.)
☐ It is a toxic comment (e.g. personal attack, insulting, etc.)

Submit

Figure 4.3: Annotation interface on Amazon Mechanical Turk.

data, this annotator will be blocked from further annotations. Each conversation is annotated by three annotators. The annotation interface is shown in Figure 4.3.

After all annotations are completed, we filter the data by removing the examples annotated by blocked workers and examples with “skip” as the majority label. The average annotation accuracy on gold data is 82%, and the Fleiss Kappa between annotators is 0.368. This Kappa score reflects a *fair agreement*. This is because the task of derailment detection can be very subjective. Therefore, we use the majority labels from the three annotators as the final label.

4.3.4 Statistics

The final MCD dataset includes 12,653 conversations with 5,269 unique images. This is because different conversations can be sourced from the same original post. Figure 4.4 shows the conversation length distribution, where the length refers to the number of context utterances plus the target text. For instances where the conversation is short, non-derailment (NON) accounts for 60% of the examples. When the conversation length exceeds three, derailment becomes the main label. This phenomenon is expected – the likelihood of a target text going off-topic increases as the conversation expands into more utterances. The longest conversation

chain in the MCD dataset contains 56 messages.

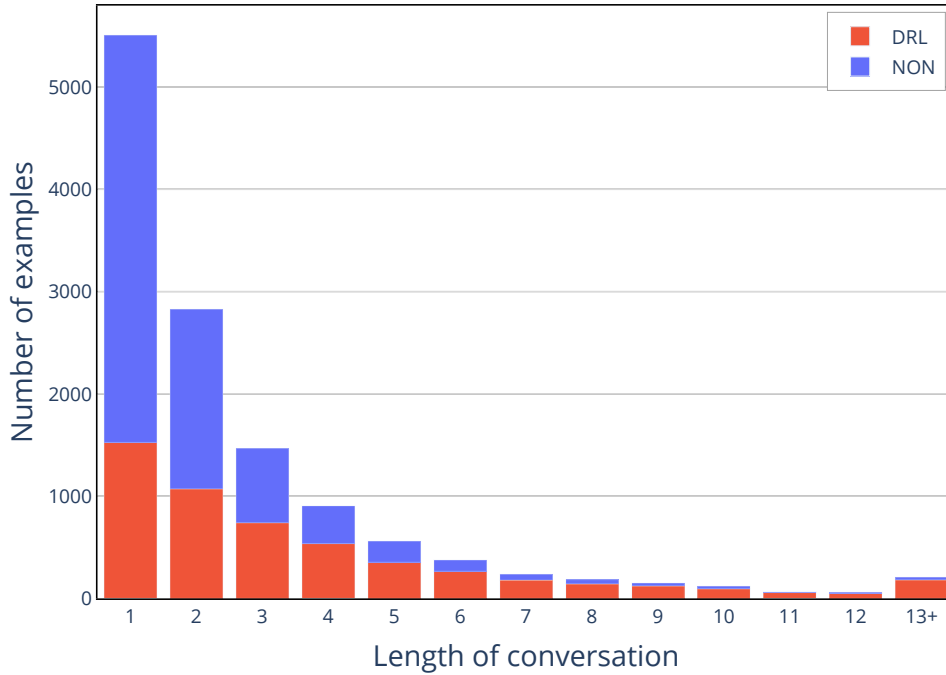


Figure 4.4: Distribution of conversation length in the MCD dataset. DRL: derailment. NON: non-derailment.

Figure 4.5 shows the distribution of fine-grained derailment reasons from the 5,313 derailment examples in the MCD dataset. Over half of all derailments are caused by irrelevancy to the title or image (or both) from the original post. The next most common reason is the introduction of new conversational topics, which contributes a further 26.3%. Taken together, irrelevancy and topic shifts constitute roughly 80 percent of all derailments, which underscores the important role of conversational context in this task. By contrast, the remaining 20% of derailments arise from toxicity and spamming content, which can potentially be detected from the target texts on their own.

Furthermore, we show the proportion of different types of natural noise in Table 4.1, with both item-level noise ratio (percentage of examples affected by a specific type of noise) and word-level noise ratio (percentage of words exhibiting a specific type of noise). Misspellings and special characters (e.g., symbols, non-ASCII/punctuation characters, etc.) are the most common source of noise. Other types of noise, such as repetitive characters, URLs, emojis, etc., are less common but still relevant in the context of Reddit posts. In addition, compared

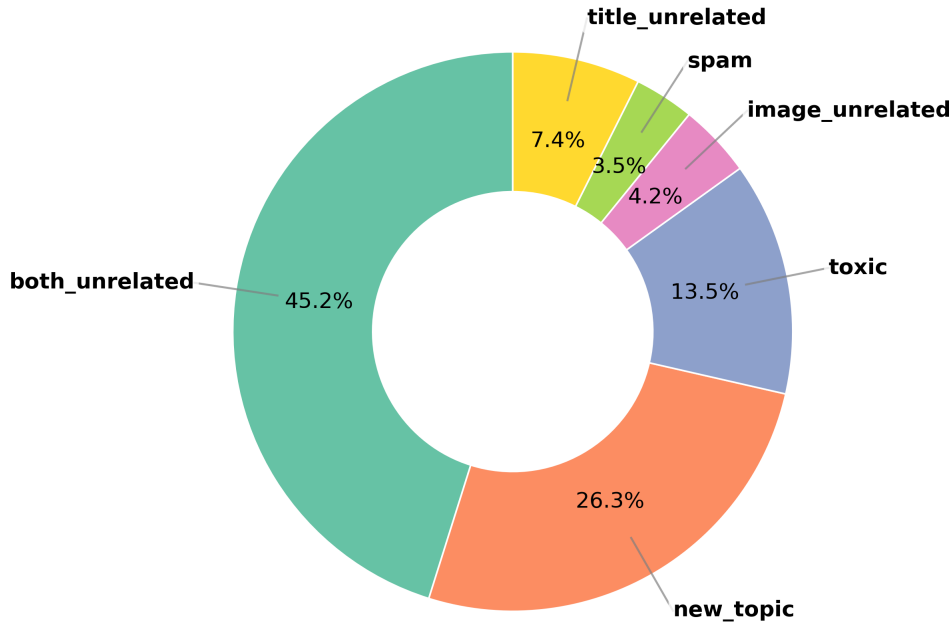


Figure 4.5: Distribution of derailment reasons in the MCD dataset.

to utterances and target text, the title is much cleaner and contains a smaller proportion of noise, which is unsurprising due to its shorter length. Overall, the dataset exhibits substantial informal and noisy user-generated content.

Noise Type	Item-level (%)				Word-level (%)			
	Title	Utterances	Text	Any	Title	Utterances	Text	Any
Misspellings	18.6	34.5	43.5	66.0	1.86	2.29	2.46	2.29
Special character	9.2	28.9	29.4	52.3	1.39	3.53	3.66	3.32
Repetitive characters	1.8	11.1	8.6	19.5	0.17	0.31	0.36	0.31
URLs	0.0	9.6	7.8	16.3	0.00	0.28	0.36	0.27
User mentions	0.2	4.9	4.1	8.5	0.02	0.12	0.15	0.12
Emojis	0.4	1.9	1.5	3.7	0.04	0.05	0.09	0.06
Hashtags	0.2	1.2	0.5	1.8	0.02	0.03	0.02	0.02

Table 4.1: Proportion of different categories of natural noise from title, utterances, text, or any of the three textual inputs in the MCD dataset.

In our experiments, we randomly split the dataset into train/dev/test sets with a ratio of 0.8/0.1/0.1. Table 4.2 shows the statistics on labels and conversation length on the data splits.

	DRL (#)	NON (#)	Avg Conv Length
Train	4,250	5,872	2.74
Dev	531	734	2.88
Test	532	734	2.82

Table 4.2: Size of the train/dev/test splits for MCD dataset. DRL: derailment. NON: non-derailment.

4.4 Model

In this section, we introduce the architecture of our proposed MMConv model, which consists of a pre-trained text encoder, a visual encoder, a context encoder, and a fusion component. Figure 4.6 gives an overview of the model architecture.

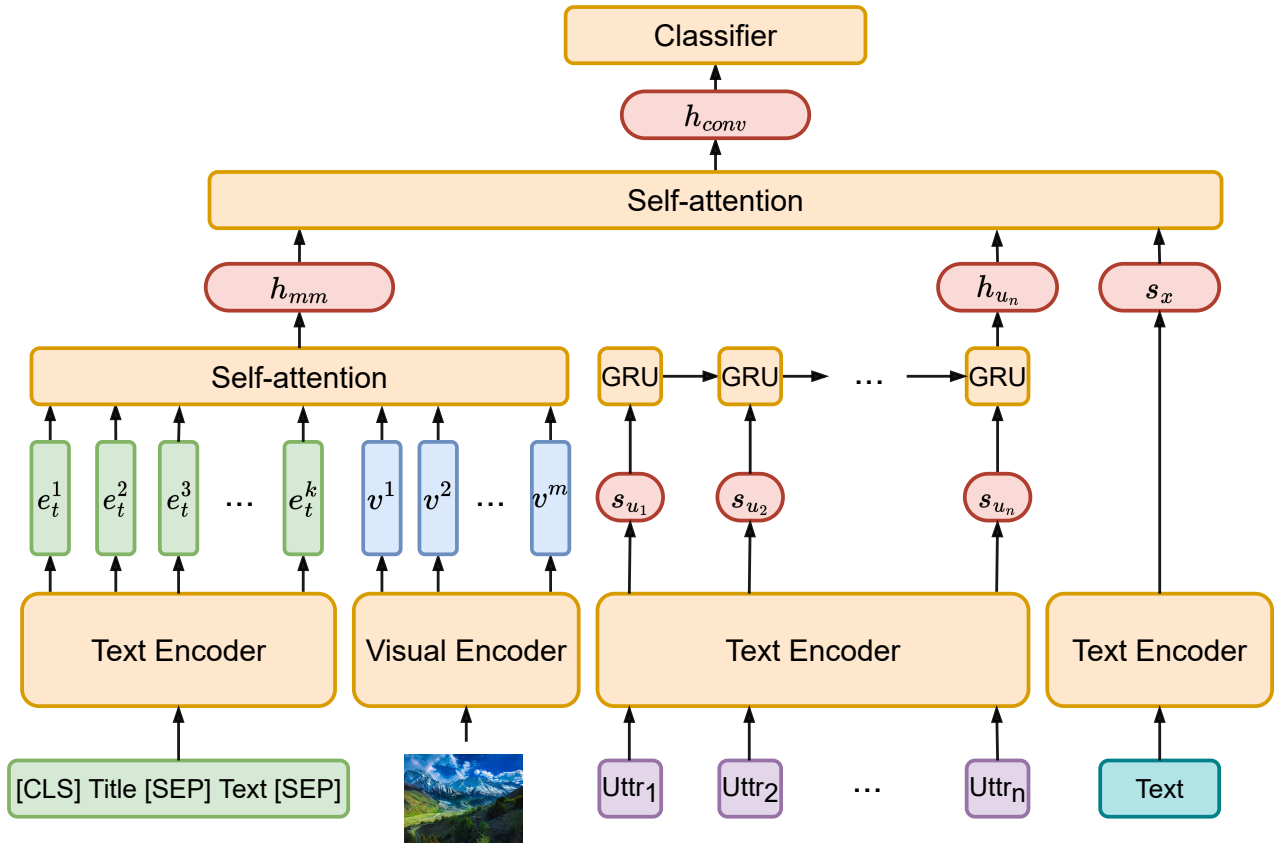


Figure 4.6: MMConv architecture.

4.4.1 Pre-trained Text Encoder

We use a pre-trained BERT (Devlin et al., 2019) model as our text encoder; however, any other pre-trained language model (e.g., RoBERTa, DeBERTa, etc.) can be used as the encoder of MMConv. Given an input text sequence $\omega = [\omega^1, \omega^2, \dots, \omega^k]$, BERT encodes the input into a sequence of vectors $\mathbf{e} = [e^1, e^2, \dots, e^k]$, each representing a token in the input text. For each utterance U_i and the target text X , we use the first hidden vector (i.e. the [CLS] token representation) as the sentence representation $s_{u_j} = e_{u_j}^1$ where $j \in \{1, 2, \dots, n\}$ and $s_x = e_x^1$.

For the title, the input is provided as “[CLS] Title [SEP] Text [SEP]”. Since the title conveys the conversation topic, we encode the title into token representations rather than sentence representations. Instead of using the hidden vector for [CLS] token, we keep the hidden representations of all tokens in the title input, thus yielding $\mathbf{e}_t$.

4.4.2 Visual Encoder

We use an image object detector as our visual encoder. Given an input image I , the visual encoder detects local regions in the image as object proposals and extracts m pooled visual features $\mathbf{O} = [o^1, o^2, \dots, o^m]$ for each object region proposal. An MLP layer with gelu activation function (Hendrycks and Gimpel, 2016) is used to map the image features into the same space as the text, thus generating the visual hidden vectors $\mathbf{V} = [v^1, v^2, \dots, v^m]$:

$$\begin{aligned} o^i &= \mathbf{Enc}_v(I), \\ v^i &= \text{gelu}(W_v o^i + b_v), \\ \forall i &\in \{1, 2, \dots, m\} \end{aligned} \tag{4.1}$$

4.4.3 Title-image Fusion

Given that both title and image form the initial topic of a conversation, we concatenate the title token vectors $\mathbf{e}_t$ with the visual hidden vectors $\mathbf{V}$ and fuse them using a self-attention

layer (Vaswani et al., 2017) to generate a multimodal title-image hidden vector h_{mm} (the title [CLS] position is used for the hidden vector):

$$h_{mm} = \mathbf{SA}([e_t^1, e_t^2, \dots, e_t^k, v^1, v^2, \dots, v^m]) \quad (4.2)$$

where k is the length of the title input and m is the number of image objects.

4.4.4 Context Encoder

To better model the chronological order in the sequence of utterances, we use a 1-layer unidirectional GRU network (Cho et al., 2014a) as our context encoder. Given the sentence representations for each utterance $[s_{u_1}, s_{u_2}, \dots, s_{u_n}]$, the context encoder iteratively computes the context hidden state h_{u_i} up to the i -th utterance:

$$h_{u_i} = \mathbf{GRU}(h_{u_{i-1}}, s_{u_i}) \quad (4.3)$$

where $i \in \{1, 2, \dots, n\}$. Therefore, the last hidden state h_{u_n} can be viewed as the encoding of the whole sequence of utterances.

4.4.5 Late Fusion

Finally, a self-attention layer is used to fuse the multimodal title-image hidden vector h_{mm} , the utterances sequence hidden vector h_{u_n} , and the target text representation s_x , generating a vector h_{conv} encoding the whole conversation. The three hidden vectors are averaged after the self-attention layer, then fed to an MLP classifier to make the final prediction:

$$\begin{aligned} h_{conv} &= \text{avg}(\mathbf{SA}([h_{mm}; h_{u_n}; s_x])) \\ p &= \text{sigmoid}(W_c h_{conv} + b_c) \end{aligned} \quad (4.4)$$

4.5 Experiments

4.5.1 Baselines

We compare to a number of text-only, conversational, and multimodal baselines:

WordOverlap (T, X)⁴ This rule-based approach defines a derailment when there is no overlapping content between the title T and the text X .

BERT-simple (T, X) (Devlin et al., 2019) This baseline uses a pre-trained BERT for classification. The title T and the text X are encoded using BERT separately, and their [CLS] token representations are concatenated and fed to a classifier head.

BERT-concat (T, U, X) This approach makes use of the sequence of utterances by concatenating the title T , utterances U , and the text X into a sequence, which is then given as input to BERT. The sequences are separated by the [SEP] token. Previous utterances are truncated when the concatenated sequence exceeds 512 tokens.

CRAFT (Chang and Danescu-Niculescu-Mizil, 2019) (T, U, X) The CRAFT model employs the HRED architecture for modeling conversations. It is first pre-trained on the contextual dialog generation task using Reddit data. When fine-tuned on the derailment task, the decoder is removed, and a classifier is attached to make the prediction.

ContextRNN (Ive et al., 2021) (T, U, X) This approach first encodes the title T , each utterance U_i in the conversation history, and the target text X separately using BERT. The [CLS] representations of the utterances are summarized into a context vector using a GRU network. Finally, the context vector, title vector, and text vector are concatenated to make the final prediction.

⁴The symbols T, I, U, X denote which types of inputs the model receives (i.e., title, image, history utterances, target text).

ToBERT (Pappagari et al., 2019) (T, U, X) Similar to ContextRNN, this approach first encodes title T , utterances U , and the text X separately. However, the context vector is modeled using an additional Transformer encoder block (Vaswani et al., 2017) with a positional embedding, taking the hidden representations of the utterances as input.

BERT-ViT (Dosovitskiy et al., 2021) (T, I, X) This approach encodes the concatenation of title T and text X (separated by [SEP] token) with BERT to get the [CLS] representations. The image I is encoded with the ViT model and is represented as a pooled visual feature vector. The joint title-text vector and the visual vector are concatenated to make the final prediction.

MMBT (Kiela et al., 2019) (T, I, X) Given that the title and image constitute the initial topic of the conversation, this approach early fuses title T and image I using BERT. The visual feature is extracted using ResNet-152 (He et al., 2016) and is concatenated with the joint title-text embedding (separated by [SEP] token) before the BERT layers.

ViLT (Kim et al., 2021) (T, I, X) The ViLT model is pre-trained on multiple vision-language tasks for better multimodal representation. The title T and the image I are encoded using the pre-trained ViLT model to get a fused vector, which is then concatenated with the text vector extracted from BERT for the final prediction.⁵

4.5.2 Pre-processing and Training

We lowercase all texts, replacing emojis and emoticons with corresponding word phrases (e.g. “=D” is replaced with “<happy>”) using the *ekphrasis* toolkit (Baziotis et al., 2017). Due to computational constraints, we only use the first and last 5 previous utterances if the length of the utterance sequence is above 10, leaving the exploration of the effect of long conversation history for future work. We extract the “bottom-up-top-down” (BUTD) (Anderson et al., 2018b)

⁵Since the pre-trained ViLT model only encodes text with less than 40 tokens, we do not concatenate the title with text using the [SEP] token.

object features for the input images with a pre-trained Faster R-CNN ResNet-101 (Ren et al., 2015) object detector. The visual features are represented with 36 object vectors $\mathbf{O} \in \mathbb{R}^{36 \times 2048}$.

We use the pre-trained BERT model (Devlin et al., 2019) (`bert-base-uncased`) as our text encoder. The model hidden size is 768, and the dropout rate (Srivastava et al., 2014) is set to 0.2. We use binary cross-entropy loss as our training loss and Adam (Kingma and Ba, 2015b) as optimizer with an initial learning rate of $3e-5$. The training batch size is 64. We train the model for 15 epochs, with early stopping when the model does not improve on the development set for 5 epochs. The best checkpoint on the development set with the highest macro-F1 score is selected for evaluation on the test set. All models are trained with 5 random seeds, and the average results are reported. All experiments are performed on an RTX A6000 GPU.

4.6 Results

4.6.1 Benchmark

We first compare the performance of the baselines and our MMConv model on the test split. The results are shown in Table 4.3, where we report the accuracy, precision, recall, F1, macro-weighted F1 (F1-mw), and the area-under-curve (AUC) score.

The results for text-only models (taking only title/text as inputs) are presented in the second and third rows. The rule-based baseline WordOverlap only achieves a performance of 0.34 F1 score, which is even lower than random guessing (0.44 F1), suggesting that the task of detecting derailment is challenging and more advanced methods are needed. The BERT-simple baseline outperforms the rule-based methods by a large margin, reaching a 0.69 F1 score on the test set.

Comparing the conversational models (taking title/utterances/text as inputs) with the text-only models, we find that the context utterances need to be handled separately. By simply concatenating all textual inputs into a whole sequence, the BERT-concat model shows a slight performance drop when utilizing the previous utterances, compared to the BERT-simple baseline. The drop might be caused by the long sequence length resulting from concatenation, and

Models	visual	conv	Acc	Pre	Rec	F1	F1-mw	AUC
Random			48.4	40.2	47.8	43.7	48.7	N/A
WordOverlap			45.5	34.6	33.8	34.2	45.2	N/A
BERT-simple			73.1	66.8	70.9	68.8	73.1	81.0
BERT-concat		✓	72.7	67.3	67.9	67.6	72.7	80.3
CRAFT		✓	74.4	70.5	67.4	68.9	74.3	81.2
CRAFT w/o pre-train		✓	71.4	66.1	65.7	65.9	71.4	78.8
ContextRNN		✓	74.7	68.5	73.0	70.7	74.7	82.7
ToBERT		✓	74.0	68.3	70.9	69.6	74.1	82.0
BERT-ViT	✓		75.0	70.5	69.7	70.1	75.0	83.7
MMBT	✓		74.9	70.5	68.9	69.7	74.8	82.4
ViLT	✓		73.5	68.4	67.6	68.0	73.3	81.7
MMConv	✓	✓	75.6	70.4	72.4	71.4	75.6	84.0
MMConv-VC	✓	✓	75.3	69.9	72.1	71.0	75.3	84.1

Table 4.3: Comparison of baselines and MMConv on the test split. *-VC: model using visual concepts instead of visual features. visual: whether the model uses images as inputs. conv: whether the model uses previous utterances as inputs.

thus, the model might have fewer attention weights for title and text tokens due to the excessive number of utterance tokens. Models that encode each utterance separately, e.g., CRAFT, ContextRNN, and ToBERT, all improve over the BERT-simple baseline, demonstrating the benefit of utilizing previous context.

It is also noticeable that CRAFT, although much smaller in model size, performs on par with BERT-based models. Gururangan et al. (2020) show that pre-training on in-domain data and a related task could result in significant benefits, therefore, we hypothesize that the improvement is from the pre-training on Reddit data (same domain) and the dialogue generation task (same conversational task). To test this hypothesis, we train CRAFT from scratch without loading the pre-trained checkpoint from Chang and Danescu-Niculescu-Mizil (2019). The performance drops by 0.03 F1 score compared to the pre-trained CRAFT, which confirms the benefits of in-domain pre-training.

Regarding the multimodal models (taking title/image/text as inputs), both BERT-ViT and MMBT show better performance than the BERT-simple baseline from using the visual information. However, the performance of ViLT is similar to that of BERT-simple, with a lower F1 score, but a higher F1-mw score. Limited by the max length of 40 tokens for text inputs, ViLT

loses crucial information when encoding the title, thus causing the performance to be worse than other multimodal counterparts.

Finally, MMConv, which takes all inputs, outperforms all other approaches, improving over BERT-simple by 0.03 F1. Inspired by Bouritsas et al. (2018), we also train a version of MMConv by replacing the visual object features with the sequence of object labels in a textual format, encoding it with BERT as the visual encoding. The model trained with visual concepts shows a similar performance to the one using object features, further validating the benefit of using visual information as additional context. By using the context utterances and the image as additional inputs, MMConv obtains higher precision, recall, and AUC scores than BERT-simple, which suggests that the improvement is not caused by the trade-off between precision and recall.

4.6.2 Ablation Analysis

To understand which components contribute the most to MMConv, we perform an ablation study by disabling one model component at a time (Table 4.4).

Models	Pre	Rec	F1	F1-mw
MMConv	70.4	72.4	71.4	75.6
w/o SA fusion	70.4	71.6	71.0	75.5
w/o context enc	70.0	69.0	69.5	74.6
w/o visual enc	72.5	66.2	69.2	75.1
w/ incongruent visual	68.9	72.0	70.4	74.7

Table 4.4: Ablation analysis for MMConv.

We first disable the self-attention fusion, where the hidden vectors of title/image, context, and main text are fused, and replace it with a concatenation of the three vectors (w/o SA fusion). This change results in a small drop in F1 while also decreasing the recall.

Removing the context/visual encoder (w/o context enc and w/o visual enc) causes a larger performance drop in F1, confirming the benefit of the additional context information sourced from the image and previous utterances. In addition, using both context and visual encoders

results in much higher recall than only using one. This indicates that the two sources of information are complementary and both provide useful information for solving the task.

Finally, to further analyze the effect of visual information, we perform an incongruent inference (Elliott, 2018), where we give as input visual features from a randomly selected image. Using incongruent visual features decreases both precision and recall, confirming that the model effectively utilizes the context from the given image.

4.6.3 Model Performance on Different Derailment Reasons

As we hypothesize in § 4.3.4, conversational models might perform better on context-related derailments (e.g., new topics, both unrelated, etc.), we analyze the model performance on different derailment reasons in Figure 4.7. The accuracy of BERT-simple (text-only), ContextRNN (conversational), and MMConv (multimodal conversational) models is compared on positive test examples with different derailment reasons. Comparing ContextRNN with BERT-simple, the introduction of conversational modeling increases the performance on derailments that require text context understanding, such as title irrelevancy or new topic introduction. After further incorporating visual information in MMConv, the performance on derailments caused by image irrelevancy can also be increased. Finally, focusing on spam content and toxic comments, the models show similar performance, which aligns with our hypothesis that these types of derailments can potentially be captured without the need for extra contexts.

4.6.4 MMConv with Different Pre-trained Text/Visual Encoders

In Table 4.5, the performance of the MMConv model with different pre-trained encoders is presented. We notice that BERT performs better than the other two pre-trained language models, i.e. RoBERTa and XLNet. Such improvement might be introduced by the next-sentence-prediction objective used in BERT pre-training, which might help conversational coherence modeling. In terms of visual encoders, BUTD features outperform ViT by a small margin.

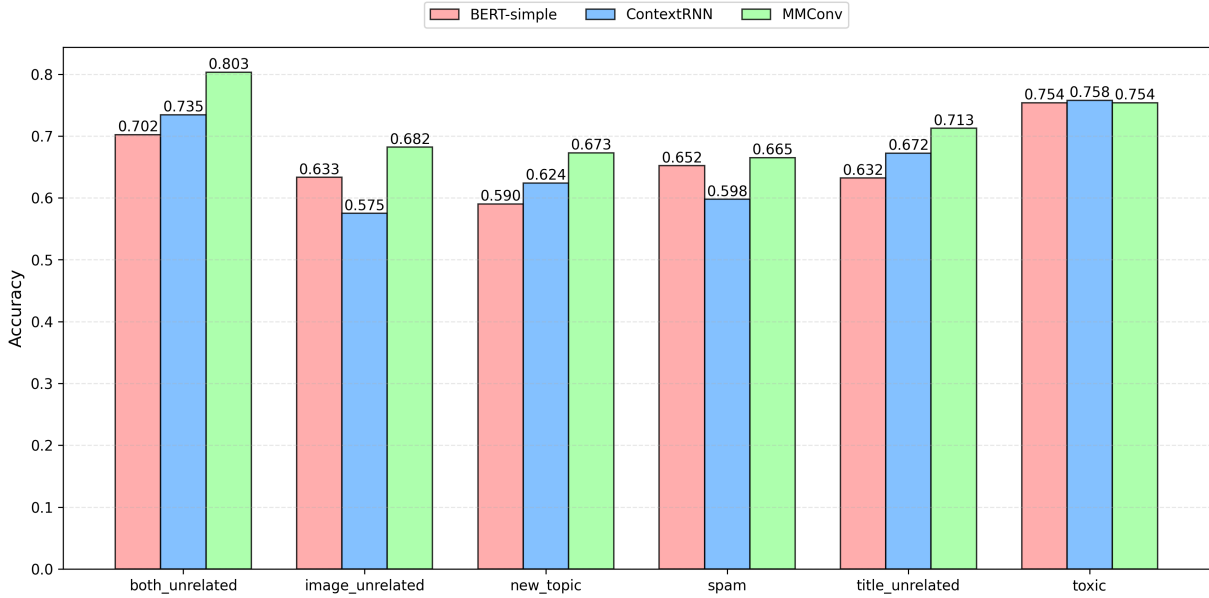


Figure 4.7: Binary accuracy on test examples with different derailment reasons.

Text	Visual	Acc	Pre	Rec	F1	F1-mw	AUC
BERT	BUTD	75.6	70.4	72.4	71.4	75.6	84.0
BERT	ViT	75.3	70.3	71.1	70.7	75.3	83.9
RoBERTa	BUTD	72.9	67.1	68.3	67.7	72.8	80.2
RoBERTa	ViT	71.5	65.8	66.2	66.0	71.4	76.7
XLNet	BUTD	72.9	65.3	75.4	70.0	73.0	81.5
XLNet	ViT	72.5	67.2	67.3	67.3	72.5	79.3

Table 4.5: Performance of MMConv model with different pre-trained text and visual encoders.

4.6.5 Forecasting Future Derailment

Zhang et al. (2018a) and Chang and Danescu-Niculescu-Mizil (2019) propose forecasting conversation derailment as a more challenging task for detecting online dialogue derailment. The task is to test if the model can predict a derailment given only historical conversation utterances. In other words, the model has no access to the actual target text and has to predict derailment based on the sequence of conversation utterances.

Following their settings, we evaluate the conversational models on the derailment examples in the test set and calculate the mean-H (average forecast horizon), which represents how many utterances ahead the model can signal a future derailment (Table 4.6). All four conversational models, although not trained specifically for forecasting derailment, can achieve a mean-H

Models	Mean-H
CRAFT	3.391
ContextRNN	3.057
ToBERT	3.247
MMConv	3.529

Table 4.6: Results of forecasting derailment for conversational models. The mean-H indicates how many utterances ahead the model can signal a derailment.

above 3 utterances, i.e., the models are capable of early warning potential derailments. CRAFT outperforms ContextRNN and ToBERT in forecasting, most likely due to its utterance-level dialogue generation pre-training, which makes the model suitable for predicting future contexts. By taking images as additional inputs, MMConv exhibits the largest mean-H.

Figure 4.8 shows the distribution of forecast horizons for the MMConv model in predicting future comments. The model does not predict future derailment on 25% of true positive examples, but could give an early signal on future derailments within 4 utterances for over 40% of all cases. Finally, the model is also capable of handling long conversations with over 13 utterances and predicting future derailment at an early stage.

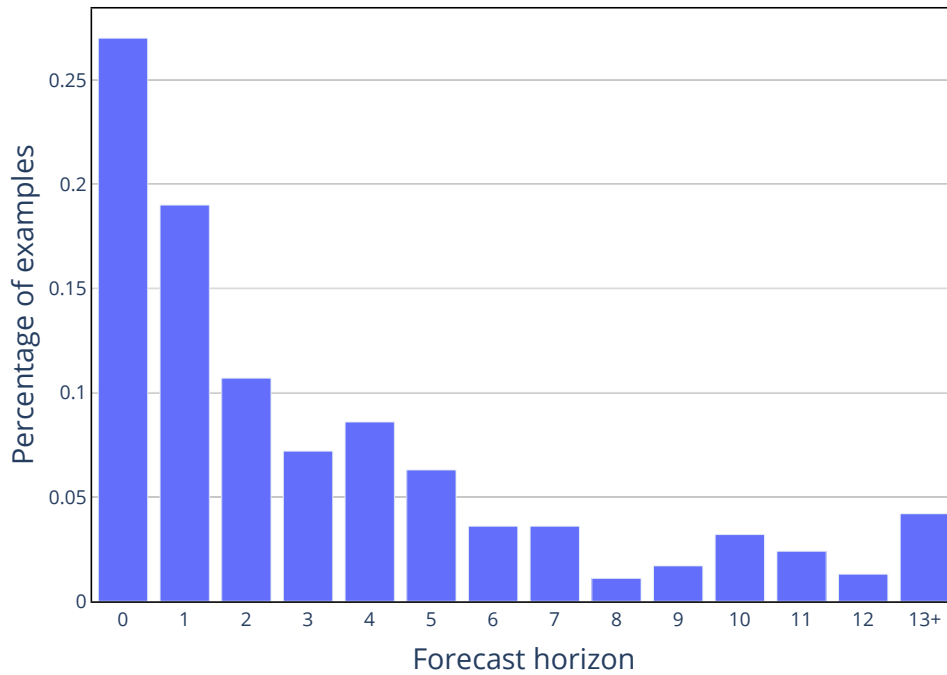


Figure 4.8: Forecast horizon distribution for MMConv model.

4.7 Robustness Analysis

4.7.1 Controlled Noise Injection

Although in Table 4.1 we show that the MCD dataset exhibits a substantial proportion of noise, the word-level noise ratio can be small and on a fixed level. Therefore, it is unknown how much improvement can be achieved from using image and conversational context under a noisier and more challenging scenario. In this section, we evaluate the models’ performance under different levels of noisy texts in a controlled style to measure their robustness. To avoid overlapping with existing noise types, we select 6 additional types of common textual noise from NL-Augmenter (Dhole et al., 2021) and apply noise injection to the test set:

- Abbreviation transformation: words are randomly replaced with their abbreviations if the abbreviations exist.
- Butter fingers perturbation: characters are randomly replaced with adjacent characters on a standard QWERTY keyboard.
- Close homophones swap: words are randomly swapped to another word that looks or sounds similar to the original word.
- Multilingual dictionary-based code switch: words are randomly replaced with their translation in another language.
- Visual attack letters: characters are randomly substituted with another character (might be from a different encoding) that looks similar.
- Whitespace perturbation: whitespaces are randomly inserted or deleted in the text.

We evaluate under two settings: noise injected into both the title and target text, and a more realistic setting where noise is injected into all textual inputs, including the title, previous utterances, and the target text. The models’ F1 scores on different proportions of noise are shown in Figure 4.9.

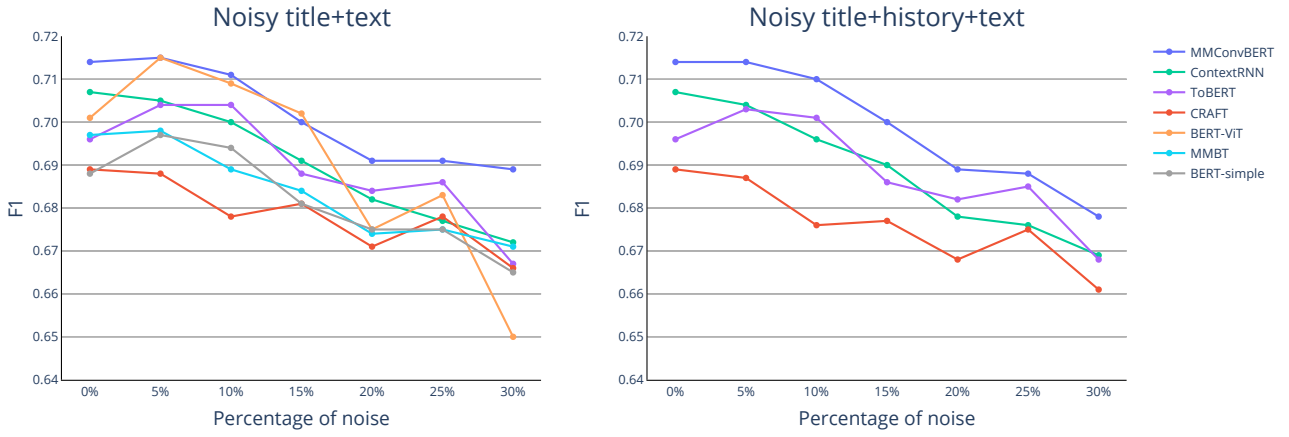


Figure 4.9: Robustness analysis on noisy test set. The model performance is evaluated on noisy title/text, and noisy title/history/text settings.

When noise is only injected into the title and text, all models show robustness against a small proportion (less than 5%). However, when the noise percentage reaches 15%, all models exhibit larger performance drops while MMConv still outperforms other baselines, which indicates that the benefits from using image and context are consistent and can help robustness.

When noise is injected into all textual inputs, similar performance drops are observed for all conversational models. Comparing the performance with and without noise injected into previous utterances, we notice that the difference within each model type is small, showing that the models are robust to the noise in previous utterances. Note that the four conversational models use a vector to encode the whole sequence of utterances rather than each utterance separately, so they are less affected by noise in individual utterances.

4.7.2 Robustness to Adversarial Attack

The other aspect of robustness is on adversarial attacks. Although the model shows robustness on noisy texts, users might escape the auto-moderation by iteratively paraphrasing the target text until the model makes a mistake, and such a process can be seen as an adversarial attack. To simulate this, we use TextAttack (Morris et al., 2020b) to perform adversarial attacks on these models. We randomly sample 200 examples from the test set and attack the model with *textfooler* attack (Jin et al., 2020). We evaluate the successful attack rate and the average number of required queries to measure the model’s robustness. If a model is robust, the attack

method would have difficulties finding an adversarial example, thus resulting in a lower successful attack rate and a higher query number. The models’ adversarial performance is presented in Table 4.7.

Models	Success rate ($\downarrow$)	Avg query ($\uparrow$)
BERT-simple	92.8	137
CRAFT	84.6	165
ContextRNN	75.2	196
ToBERT	79.6	186
MMConv	80.2	188

Table 4.7: Results of successful attack rate and average query number with textfooler attack (Jin et al., 2020).

The BERT-simple baseline exhibits the least robustness to attack, with a successful attack rate of 92.8% and an average query number of 137. Other conversational models all show obvious improvements over the text-only baseline, indicating that the additional conversational context could increase model robustness. Among all models, ContextRNN shows the best robustness, decreasing the successful attack rate significantly to 75.2%, and increasing the average query number to 196. Even though using conversational context results in significant improvement in success rate, the successful attack rate of 75% can be too high for real-world applications, leaving the robustness to adversarial attack an important direction to investigate.

4.8 Conclusions

In this chapter, we explore the task of detecting online conversation derailment. We take a broader view of derailment and focus on conversation derailment based on topic changes. The main contribution is the creation of a new multimodal Conversation Derailment (MCD) dataset, which introduces multimodality into conversation derailment detection, as well as including naturally occurring textual noise created by humans. Furthermore, we propose a novel hierarchical architecture – MMConv – that uses visual and conversational contexts. Our experiments show that the proposed model outperforms strong baselines in conversation derailment detection on the naturally noisy text data. Last but not least, we perform a controlled robustness analysis

by evaluating models with synthetically noisier texts. We find that models using visual and conversational information have increased robustness, which can help build more accurate and practical moderation applications. The experimentation results help answer research question 1 and further validate the positive effect of incorporating additional contextual information to mitigate induced vulnerabilities arising from textual noise. However, despite the increased robustness to noise, models with conversational context still show a poor robustness to adversarial attacks, leaving the defense against adversarial attacks an important direction to explore. In the next chapter, we will move from the challenge of addressing the single-turn natural noise to dealing with adaptive perturbations – adversarial attacks – by iteratively removing noise from the model’s hidden representations.

Chapter 5

Improved Robustness to Adversarial Attacks via Iterative Denoising

In Chapter 4, we demonstrated that even though the model’s robustness to noise has been improved by employing extra information from images or conversational contexts, it can still be vulnerable to adversarial attacks. In this chapter, we focus on defending against textual adversarial attacks. Pre-trained language models have significantly advanced performance across various natural language processing tasks. Nevertheless, adversarial attacks continue to pose a critical challenge to systems built using these models, as they can be exploited with carefully crafted adversarial texts.

Chapter 3 integrates an error-correction approach that cleans the noisy input with an encoder-decoder model. This method shows promising results in purifying textual noise. In this chapter, following the strategy of removing noise, we apply the noise removal in the latent space for better generalization and denoise the model’s hidden states to purify the adversarial perturbations. Inspired by the ability of diffusion models to predict and reduce noise in computer vision, we propose a novel and flexible adversarial defense method for language classification tasks, *DiffuseDef*, which incorporates a diffusion layer as a denoiser between the encoder and the classifier. The diffusion layer is trained on top of the existing classifier, ensuring seamless integration with any model in a plug-and-play manner. During inference, the adversarial hidden

state is first combined with sampled noise, then denoised iteratively and finally ensembled to produce a robust text representation. By integrating adversarial training, denoising, and ensembling techniques, we show that DiffuseDef improves over existing adversarial defense methods and achieves state-of-the-art performance against common black-box and white-box adversarial attacks.

This chapter is based on the paper “DiffuseDef: Improved Robustness to Adversarial Attacks via Diffusion Training via Iterative Denoising” in *ACL, 2025* (Li et al., 2024). The codebase is available at <https://github.com/Nickeilf/DiffuseDef>.

5.1 Introduction

Pre-trained language models (PLM) have significantly advanced the performance of various natural language processing (NLP) tasks. Despite such improvements, current NLP systems remain susceptible to adversarial attacks where carefully crafted text perturbations can lead to incorrect model outputs (Alzantot et al., 2018; Jin et al., 2020; Li et al., 2020a). In order to improve robustness to adversarial attacks, various defense methods have been proposed, such as adversarial training (Zhu et al., 2020; Si et al., 2021; Zhou et al., 2021c; Xi et al., 2022), text denoising (Nguyen Minh and Luu, 2022; Wang et al., 2023), ensembling (Zhou et al., 2021c; Zeng et al., 2023; Li et al., 2023), etc. However, existing defense methods either assume the test-time perturbation/attack set is similar to that used in training (Li et al., 2021b), or are limited to specific architectures (Xi et al., 2022), or at inference time require large computational costs, thereby limiting their practical applicability.

In this work, inspired by the general noise prediction and reduction capability of diffusion models, we propose *DiffuseDef*, a novel adversarial defense method that employs diffusion training to denoise hidden representations of adversarial texts. Unlike Li et al. (2023) and Yuan et al. (2024), which apply diffusion on texts or labels, DiffuseDef directly removes noise from the hidden states, providing a more effective and robust text representation to defend against adversarial texts. Different from diffusion-based defenses in CV (Nie et al., 2022; Gulsen et al.,

2024) that apply purification to input images, DiffuseDef only performs denoising at the final hidden layer, resulting in an improvement in efficiency. This improvement is further enhanced when combined with ensembling, as it eliminates the need for a forward pass through all classifier parameters for each ensemble.

DiffuseDef combines adversarial training with diffusion training, where the diffusion layer is trained to predict randomly sampled noise at a given timestep. During inference, the diffusion layer serves as a denoiser, iteratively removing noise from adversarial hidden states to yield a robust hidden representation. Moreover, we adopt the ensembling strategy by first adding random noise to the text hidden states to create multiple variants, then denoising them via the diffusion layer. The model output is made by averaging all denoised hidden states. Since ensembling happens solely at the diffusion layer, DiffuseDef is more efficient than traditional ensembling-based methods (Ye et al., 2020; Zeng et al., 2023), which require a full forward pass through all model parameters.

Through systematic experimentation, we demonstrate that DiffuseDef outperforms strong defense methods and can defend against multiple types of black-box and white-box adversarial attacks, while preserving performance on clean texts. Our analysis also reveals that the ensembling of diffused representations provides a stronger defense against finding vulnerable words to attack and can reduce the distance in latent space between adversarial texts and their clean text counterparts.

In this work, our contributions can be summarized as follows:

- We propose DiffuseDef, a novel and flexible adversarial defense method that can be added on top of any existing adversarial defense methods to further improve robustness to adversarial attacks.
- DiffuseDef outperforms existing adversarial methods and achieves state-of-the-art performance against prevalent black-box and white-box attacks.
- Through extensive analysis, we demonstrate the effectiveness of the ensembling diffused representation and the efficiency of DiffuseDef compared to existing ensembling-based

methods.

5.2 Related Work

Section 2.2.2.3 categorizes textual adversarial defenses into adversarial training, denoising, ensembling and robust certification. In this section, we provide more details on the relevant work that forms the basis of our proposed DiffuseDef.

Adversarial training Research in textual adversarial defense initially started with the vision-inspired idea of adversarial training, adding small perturbations to word embeddings so the model learns to be locally invariant (Miyato et al., 2016). The basic recipe soon evolved: FreeLB++ (Li et al., 2021b) inject adversarial gradients to the embedding and performs several gradient ascent in a training batch; SMART (Jiang et al., 2020) introduces adversarial regularization during the training to prevent aggressive perturbation; and later variant like RSMT (Moon et al., 2023) combines adversarial noise and random smoothing to adapt model to noisy representation. Although these methods harden models against the specific perturbation patterns seen in training, their protection typically wanes when the attacker changes tactics or crafts unseen manipulations, hence the need for complementary defenses.

Diffusion denoising Diffusion models are commonly used in computer vision (CV) to generate high-quality images by predicting and removing noise from a sampled noisy image. Therefore, they can be adopted to remove noise from adversarial images and thus improve robustness to attacks (Im et al., 2016; Nie et al., 2022; Gulsen et al., 2024). However, in NLP, minimal research has investigated adversarial defense with diffusion models due to the discrete and contextual nature of text data. Li et al. (2023) adopt the idea of iterative denoising and reconstruct adversarial texts from masked texts, while Yuan et al. (2024) use a diffusion model as a classifier and perform reverse diffusion steps on the label vector, conditioning on the input text. In this work, we apply the diffusion layer to DiffuseDef and use it to iteratively remove noise from the adversarial hidden states.

5.3 DiffuseDef

5.3.1 Training

The proposed diffusion defense model consists of a pre-trained encoder for feature extraction, a transformer-based diffusion layer for noise prediction and reduction, and a classifier layer for output generation. The training process is divided into two stages: **adversarial training** and **diffusion training** (Figure 5.1). The **adversarial training** stage employs any neural network-based adversarial training methods such as FreeLB++ (Li et al., 2021b) and RSMI (Moon et al., 2023). For each minibatch in the training set, a loop of several gradient-ascent steps is performed, and small, learnable perturbations are added to the input embeddings at each gradient ascent step. Therefore, the model will be trained on embeddings with adversarial noise, which aims to maximize the loss and thus achieve improved adversarial robustness to text examples. In this stage, the encoder and the classifier are trained with the adversarial task loss.

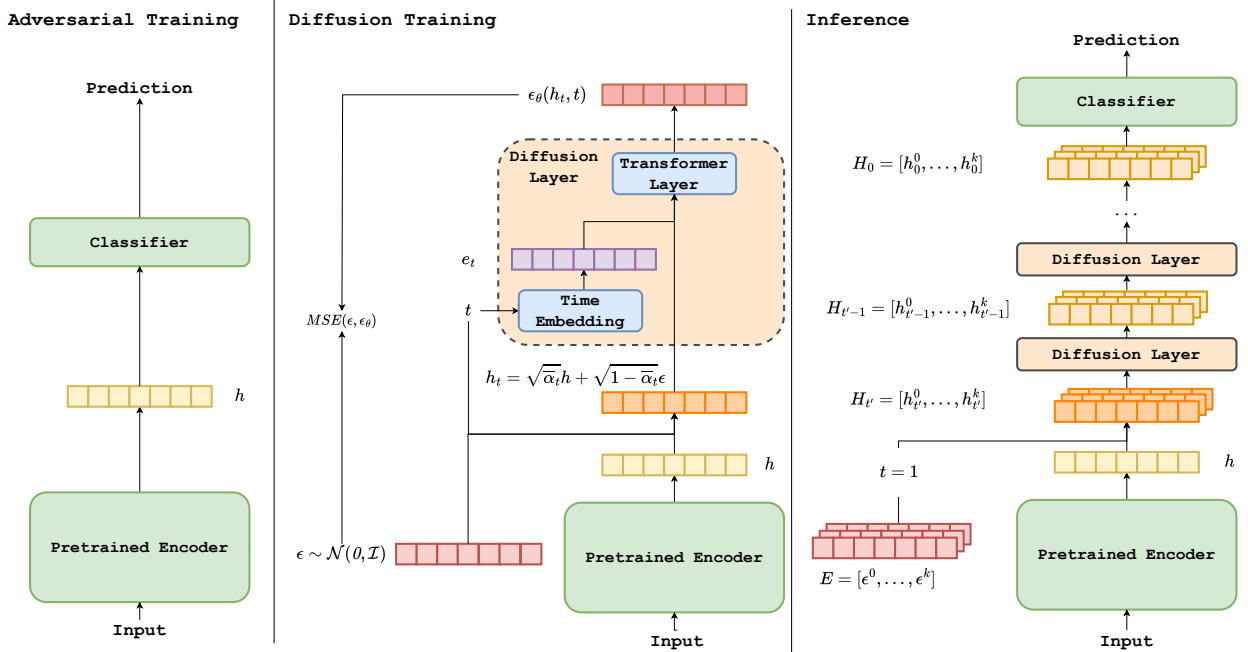


Figure 5.1: Training and inference of DiffuseDef model. The *adversarial training* stage trains the pre-trained encoder and classifier with perturbed input for adversarial robustness. The *diffusion training* stage trains the diffusion layer to predict noise injected at a given timestep t . At *inference* time, the text hidden state is first noised by one step and then denoised by t' steps to create the denoised hidden states, which are ensembled to produce the output.

In the **diffusion training** stage, only the diffusion layer is trained to predict random noise added to the clean text hidden state at different timesteps. This enables the model to denoise the adversarial hidden state at inference time. The pre-trained encoder, however, is frozen during this stage. Since the pre-trained encoder is only used for feature extraction, the diffusion training method is compatible with any neural network-based adversarial training method, making it highly flexible.

Given an input sequence of tokens $\mathbf{x} \in \mathbb{R}^L$, the pre-trained encoder extracts a hidden state $h \in \mathbb{R}^{L \times D}$. A random Gaussian noise ϵ is sampled to perturb the hidden state h . The forward diffusion process, as defined by Sohl-Dickstein et al. (2015), is a Markov Chain where at each timestep a Gaussian noise is sampled and added to the previous latent feature:

$$h_t = \sqrt{1 - \beta_t} h_{t-1} + \sqrt{\beta_t} \epsilon \quad (5.1)$$

where $\epsilon \in \mathcal{N}(0, \mathcal{I})$, h_t is the noisy hidden state at step t and β is a pre-calculated variance schedule changing with t .

As shown by Ho et al. (2020), this equation can be reformulated to calculate h_t directly from h by defining $\alpha_t = 1 - \beta_t$ and $\bar{\alpha} = \prod_{i=1}^t \alpha_i$, thus

$$h_t = \sqrt{\bar{\alpha}_t} h + \sqrt{1 - \bar{\alpha}_t} \epsilon \quad (5.2)$$

At each training step, a random forward diffusion timestep t is sampled from a uniform distribution. Therefore, the noisy hidden state h_t is created from h , t , and ϵ . The diffusion layer θ consists of a time embedding and a transformer layer. The time embedding receives the diffusion timestep t as input and produces an embedding e_t , which is added to h_t as input for the transformer layer. Finally, the transformer layer outputs the predicted noise $\epsilon_\theta(h_t, t)$, and a mean square error is used to compute the loss between the predicted noise $\epsilon_\theta(h_t, t)$ and the actual sampled noise ϵ .

$$L = \mathbb{E}_{t, h, \epsilon} \left[\left\| \epsilon - \epsilon_\theta(\sqrt{\bar{\alpha}_t} h + \sqrt{1 - \bar{\alpha}_t} \epsilon) \right\|^2 \right] \quad (5.3)$$

5.3.2 Inference

Leveraging the diffusion layer’s ability to predict noise at a given timestep t , we adopt it as a denoiser during inference by iteratively performing the reverse diffusion steps, which sample from $p_\theta(h_{t-1}|h_t) = \mathcal{N}(h_{t-1}; \mu_\theta(h_t, t), \Sigma_\theta(h_t, t))$ to produce the denoised hidden state

$$\mu_\theta(h_t, t) = \frac{1}{\sqrt{\alpha_t}} \left(h_t - \frac{1 - \alpha_t}{\sqrt{1 - \bar{\alpha}_t}} \epsilon_t \right) \quad (5.4)$$

$$\Sigma_\theta(h_t, t) = \sigma_t^2 \mathcal{I} \quad (5.5)$$

where ϵ_t is the predicted noise from the diffusion layer and $\sigma_t^2 = \beta_t$. The denoised hidden state can thus be computed with

$$h_{t-1} = \frac{1}{\sqrt{\alpha_t}} \left(h_t - \frac{1 - \alpha_t}{\sqrt{1 - \bar{\alpha}_t}} \epsilon_t \right) + \sigma_t z \quad (5.6)$$

where $z \in \mathcal{N}(0, \mathcal{I})$.

Inference in DiffuseDef consists of three key steps: a one-step **noising**, a multi-step **denoising**, and a final **ensembling** step. After the pre-trained encoder extracts its hidden state h , a set of k Gaussian noise vectors $E = [\epsilon^0, \epsilon^1, \dots, \epsilon^k]$ is sampled to perform a single forward diffusion step. These noise vectors E are then added to the hidden state h following equation 5.2, resulting in a set of noisy hidden states $H_{t'} = [h_{t'}^0, h_{t'}^1, \dots, h_{t'}^k]$, where t' denotes the number of denoising steps. This noise-adding step enables the possibility of ensembling by sampling from Gaussian noise. It also makes the ensembling more efficient as each ensemble only requires an additional forward pass over the diffusion layer rather than the full model parameters. The noisy hidden states $H_{t'}$ are subsequently denoised through t' reverse diffusion steps, where noise is predicted by the diffusion layer and subtracted from the previous noisy hidden states. Unlike Ho et al. (2020) where the reverse diffusion step starts with pure noise sampled from a standard normal distribution, we assume the noisy hidden state $H_{t'}$ is already an intermediate state in the reverse diffusion steps. This allows us to use a smaller number of t' than the training timestep t to prevent the denoised hidden states from diverging substantially from the initial hidden state h . This sequence of denoising steps creates the final denoised hidden states $H_0 = [h_0^0, h_0^1, \dots, h_0^k]$,

which are averaged and used by the classifier to output the final predicted label. This process is summarized in Algorithm 1.

Algorithm 1: Inference of DiffuseDef

Data: Input text $\mathbf{x}$
Result: Predicted label y'

- 1 $h \leftarrow \text{Enc}(\mathbf{x});$
- 2 Sample $E = [\epsilon^0, \epsilon^1, \dots, \epsilon^k], \epsilon \sim \mathcal{N}(0, \mathcal{I});$
- 3 $H_{t'} \leftarrow \sqrt{\bar{\alpha}_1}h + \sqrt{1 - \bar{\alpha}_1}E;$
- 4 **for** $i \leftarrow 0$ **to** $t' - 1$ **do**
- 5 $E_{t'-i} \leftarrow \epsilon_\theta(H_{t'-i}, t' - i);$
- 6 $H_{t'-i-1} \leftarrow \frac{1}{\sqrt{\alpha_{t'-i}}} \left(H_{t'-i} - \frac{1 - \alpha_{t'-i}}{\sqrt{1 - \bar{\alpha}_{t'-i}}} E_{t'-i} \right) + \sigma_{t'-i}z;$
- 7 **end**
- 8 $h_{\text{ensemble}} \leftarrow \text{avg}(H_0);$
- 9 $y' \leftarrow \text{CLS}(h_{\text{ensemble}});$

5.4 Experiments

Datasets We focus on two NLP tasks in our experiments: topic classification and natural language inference (NLI), and conduct experiments on three common datasets: AG News (Zhang et al., 2015), IMDB (Maas et al., 2011), and QNLI datasets (Wang et al., 2018). We randomly split AGNews, IMDB, and QNLI datasets into train, validation, and test splits.

Dataset	Train	Valid	Test	Avg Len
AGNews	108K	12K	7K	51.3
IMDB	40K	5K	5K	311.9
QNLI	94K	10K	5K	47.2

Table 5.1: Dataset statistics. The average text length is counted with BertTokenizer.

Table 5.1 presents the number of examples in train/valid/test splits and the average token length for the three datasets used in the experiments. For QNLI and AGNews datasets, we randomly split the training set into our train/valid splits, with a ratio of 0.9/0.1, and use their test sets as our test splits. For the IMDB dataset, we randomly split the dataset into train/valid/test splits with a ratio of 0.8/0.1/0.1. All train/valid/test splitting is performed using a random seed of 42.

Evaluation Following previous work on adversarial defense, we use three benchmarking black-box attack methods to evaluate the robustness of DiffuseDef: TextFooler (TF) (Jin et al., 2020), TextBugger (TB) (Li et al., 2019a), and Bert-Attack (BA) (Li et al., 2020a). We also evaluated the robustness against two white-box attacks: T-PGD (Yuan et al., 2023) and SemAttack (Wang et al., 2022a). Regarding evaluation metrics, we measure the clean accuracy (**Clean%**) on the test set, the accuracy under attack (**AUA%**), and the number of adversarial queries (**#Query**) needed for a successful attack. Higher scores on the three metrics denote a better robustness performance of a defense method. The accuracy on clean data is measured across the entire test set. The accuracy under attack and number of queries, due to the lengthy attacking process, is calculated on a randomly sampled subset of 1000 examples from the test set. The evaluation metrics are averaged based on experiments run with 5 random seeds. We use the **TextAttack** (Morris et al., 2020b) library as the adversarial evaluation framework. To ensure a fair comparison and high-quality adversarial examples, we follow the same evaluation constraints as in Li et al. (2021b).

Dataset	$\varepsilon_{\min}$	$\mathbf{K}_{\max}$	$\rho_{\max}$
AGNews	0.84	50	0.3
IMDB	0.84	50	0.1
QNLI	0.84	50	0.2

Table 5.2: Evaluation parameters for each dataset.

As shown in Table 5.2, when evaluating with adversarial attack, we follow the parameter settings for TextAttack as suggested by Li et al. (2021b). The minimum semantic similarity $\varepsilon_{\min}$ between the clean text and adversarial text is set to 0.84, with the score computed using Universal Sentence Encoder (Cer et al., 2018). The maximum number of candidate substitutions $\mathbf{K}_{\max}$ from the attacker is 50, thus, the maximum number of queries $\mathbf{Q}_{\max} = \mathbf{K}_{\max} \times \mathbf{L}$ where $\mathbf{L}$ is the number of tokens. Finally, the maximum percentage of changed tokens $\rho_{\max}$ is set to 0.3/0.1/0.2 for AGNews, IMDB, and QNLI datasets, respectively.

Comparison to SOTA We compare our proposed method with state-of-the-art adversarial defense approaches, trained using both BERT (Devlin et al., 2019) and RoBERTa (Liu et al., 2019b) as backbones:

- **Fine-tune**: Fine-tuning pre-trained models on downstream tasks with no defense method applied. This is a baseline approach used to illustrate the effect of adversarial attacks.
- **InfoBERT** (Wang et al., 2021): Applying mutual-information-based regularizers during fine-tuning of pre-trained models to improve robustness.
- **FreeLB++** (Li et al., 2021b): An adversarial training method improving on FreeLB (Zhu et al., 2020), which adds adversarial perturbations to word embedding during fine-tuning.
- **EarlyRobust**¹(Xi et al., 2022): Extracting early-bird subnetworks and pruning pre-trained models for efficient adversarial training.
- **RMLM** (Wang et al., 2023): A denoising-based model combined with adversarial text detection.
- **ATINTER** (Gupta et al., 2023): A fine-tuned T5 model that rewrites the input adversarial texts, and the classifier predicts labels from the rewritten texts.
- **RSMI** (Moon et al., 2023): A two-stage training method that combines randomized smoothing and masked inference to improve adversarial robustness.

Implementation and Settings We train two DiffuseDef variants using FreeLB++ and RSMI models as base models, considering their robust adversarial defense capabilities. In the diffusion layer, only one transformer encoder layer (Vaswani et al., 2017) is used for efficiency purposes. The maximum noising timestep t during training is set to 30 for AGNews and QNLI datasets, and 10 for the IMDB dataset, while at inference time, we only apply 5 denoising steps for t' . We follow (Ho et al., 2020) to use a linear β_t schedule from $\beta_1 = 10^{-4}$ to $\beta_t = 0.02$. The base encoder and classifier are first fine-tuned for 10 epochs, and then the diffusion layer is trained from scratch for 100 epochs, with the base encoder and classifier parameters frozen for efficiency. During the diffusion training stage, the same train-dev splits are used as in the adversarial training stage, thus ensuring no data leakage. At inference time, the number of ensembles is set to 10.

¹We only run EarlyRobust with BERT as its implementation with RoBERTa has not been released.

	AGNews	IMDB	QNLI
Epochs	100	100	100
Batch size	64	64	64
Sequence len	128	256	256
Dropout	0.1	0.1	0.1
Optimizer	AdamW	AdamW	AdamW
Lr	2e-5	2e-5	2e-5
t	30	10	30
t'	5	5	5
k	10	10	10

Table 5.3: Hyperparameters for diffusion training.

The details on hyperparameters of diffusion training can be found in Table 5.3. The adversarial training, i.e., FreeLB++ and RSMT, follows the hyperparameter settings from the corresponding papers. All models are trained on a single RTX A6000 GPU. The diffusion training of 100 epochs takes 6/4/3 hours on AGNews, IMDB, and QNLI datasets, respectively.

5.5 Results and Analysis

5.5.1 Adversarial Robustness

In Table 5.4, we compare the adversarial robustness of DiffuseDef with baselines and SOTA methods on AGNews, IMDB, and QNLI datasets trained with BERT and RoBERTa. DiffuseDef consistently outperforms all other methods on all three datasets across both PLMs and the three attacks, exhibiting substantial improvements in accuracy under attack. After applying diffusion training, the AUA score for both FreeLB++ and RSMT models improves significantly, with an average increase of 30% AUA against the three attack methods compared to without applying diffusion training. Note that despite the robust adversarial performance of the RSMT model, especially when trained with RoBERTa on the IMDB dataset, it still benefits from DiffuseDef. When comparing the clean accuracies to its base model (i.e., FreeLB++ and RSMT), DiffuseDef only shows a minor decline, between 0.2% and 0.7% accuracy score, which indicates that it can preserve the clean text performance while improving adversarial robustness. Moreover, models

Dataset	PLM	Method	Clean%	AUA%			#Query		
				TF	TB	BA	TF	TB	BA
AGNews	BERT-base	Fine-Tuned	94.4	10.2	25.4	27.1	348	372	379
		InfoBERT (Wang et al., 2021)	95.0	35.5	39.1	42.6	377	397	397
		FreeLB++ (Li et al., 2021b)	95.0	54.7	56.5	44.6	426	430	390
		EarlyRobust (Xi et al., 2022)	94.4	35.6	37.2	45.7	475	516	533
		RMLM (Wang et al., 2023)	94.3	45.6	44.4	54.1	642	738	705
		ATINTER (Gupta et al., 2023)	94.2	68.0	59.0	81.0	527	235	122
		RSMI (Moon et al., 2023)	94.3	52.6	56.7	55.4	680	737	687
		DiffuseDef-FreeLB++ (Ours)	94.8	84.5	86.0	84.6	877	972	910
		DiffuseDef-RSMI (Ours)	93.8	82.7	83.3	84.4	894	1029	930
	RoBERTa-base	Fine-Tuned	94.9	34.1	36.9	43.6	372	396	410
		InfoBERT (Wang et al., 2021)	95.5	40.2	45.2	48.6	392	421	430
		FreeLB++ (Li et al., 2021b)	95.4	57.5	62.9	55.9	444	467	447
		RSMI (Moon et al., 2023)	93.1	64.2	66.4	67.4	774	861	808
		DiffuseDef-FreeLB++ (Ours)	95.3	85.6	87.6	85.3	880	976	906
		DiffuseDef-RSMI (Ours)	92.9	82.9	83.5	82.2	905	925	1047
IMDB	BERT-base	Fine-Tuned	93.3	7.7	8.3	10.5	540	534	378
		InfoBERT (Wang et al., 2021)	93.9	29.2	25.4	30.7	642	644	390
		FreeLB++ (Li et al., 2021b)	94.3	44.2	39.6	40.6	784	829	426
		EarlyRobust (Xi et al., 2022)	92.7	49.7	46.8	43.8	2267	2788	1841
		RMLM (Wang et al., 2023)	93.6	48.8	47.9	47.1	2554	3258	2064
		ATINTER (Gupta et al., 2023)	94.3	25.0	15.0	39.0	1173	645	637
		RSMI (Moon et al., 2023)	90.9	60.0	54.4	51.1	2840	3455	2070
		DiffuseDef-FreeLB++ (Ours)	94.4	82.1	83.0	84.0	3174	4348	2842
		DiffuseDef-RSMI (Ours)	90.2	80.9	79.8	79.8	3590	4748	2901
	RoBERTa-base	Fine-Tuned	94.6	21.3	17.9	13.6	587	671	493
		InfoBERT (Wang et al., 2021)	94.8	30.9	27.9	21.8	681	760	549
		FreeLB++ (Li et al., 2021b)	95.3	46.0	42.1	33.9	829	974	637
		RSMI (Moon et al., 2023)	92.7	77.9	74.3	70.6	3443	4342	2619
		DiffuseDef-FreeLB++ (Ours)	95.0	86.2	85.9	86.8	3573	4663	2941
		DiffuseDef-RSMI (Ours)	92.4	84.7	84.1	84.3	3673	4782	3007
QNLI	BERT-base	Fine-Tuned	90.8	21.5	15.5	13.3	195	206	177
		InfoBERT (Wang et al., 2021)	91.2	27.8	22.8	18.7	217	232	201
		FreeLB++ (Li et al., 2021b)	90.3	45.6	40.2	30.5	253	279	226
		EarlyRobust (Xi et al., 2022)	89.2	24.3	21.2	19.1	265	292	255
		ATINTER (Gupta et al., 2023)	90.4	43.0	24.8	34.0	185	115	89
		RSMI (Moon et al., 2023)	87.4	35.2	30.9	28.2	314	353	314
		DiffuseDef-FreeLB++ (Ours)	90.3	66.7	65.3	64.4	485	587	543
		DiffuseDef-RSMI (Ours)	86.4	55.5	54.8	57.2	459	569	528
	RoBERTa-base	Fine-Tuned	92.8	26.6	22.5	20.3	204	219	188
		InfoBERT (Wang et al., 2021)	92.5	29.1	25.8	20.3	205	223	189
		FreeLB++ (Li et al., 2021b)	92.8	33.9	27.7	20	227	244	200
		RSMI (Moon et al., 2023)	89.3	38.8	33.2	30.0	340	375	343
		DiffuseDef-FreeLB++ (Ours)	92.7	64.6	64.3	61.5	473	579	524
		DiffuseDef-RSMI (Ours)	88.8	57.7	55.7	53.9	469	578	518

Table 5.4: Black-box adversarial robustness results on classification tasks with BERT and RoBERTa. *Clean*: accuracy on clean test set. *TF*: TextFooler. *TB*: TextBugger. *BA*: BertAttack.

trained with DiffuseDef show a much smaller gap between clean accuracy and accuracy under attack, and such a difference can be reduced to less than 10% AUA.

Another benefit of DiffuseDef is the increased number of adversarial queries needed to obtain a successful attack. Models applying DiffuseDef require over twice the number of queries on all three datasets compared to the other methods. This increase is even larger on the IMDB dataset due to the longer text length. For example, the DiffuseDef model requires over 3000 queries to achieve a successful attack, while FreeLB++ only needs 400 to 800 queries. The significant increase suggests that even if the attackers manage to construct a successful adversarial attack, they need 2x to 3x more time to find the attack on DiffuseDef than other models, affirming the improved robustness from diffusion training. Moreover, we observe that the number of queries for denoising-based methods (i.e., RSMI, DiffuseDef) is generally higher than adversarial training-based methods (i.e., InfoBERT, FreeLB++). This is because denoising-based methods transform the hidden representations of the adversarial texts into a non-deterministic representation. The introduction of randomness in hidden states results in uncertainty in model logits, thus increasing the difficulty of finding vulnerable words to attack (Wang et al., 2023).

5.5.2 Robustness against White-box Attacks

In addition to black-box attacks, we also measure the robustness against white-box attacks, where the attackers can access the defender model’s gradients and parameters, which is a more challenging adversarial attack scenario than black-box attacks. In Table 5.5, we report AUA score under T-PGD (Yuan et al., 2023) and SemAttack (Wang et al., 2022a), the former being a variant of PGD attack (Madry et al., 2019) for text, while the latter adopts C&W attack (Carlini and Wagner, 2017) on embedding layers. We observe that white-box attacks result in lower AUA for all defense methods. For example, FreeLB++ performs poorly on IMDB when attacked by SemAttack, with an AUA score being 0.03. However, applying diffusion training still significantly improves adversarial robustness to white-box attacks. Note that even RSMI shows very robust performance against T-PGD², it can still further improve from DiffuseDef.

²RSMI is incompatible with SemAttack because it requires gradients of the embedding layer, but SemAttack skips the embedding layer by sending embeddings directly to the target model.

Method	AGNews		IMDB	
	T-PGD	SemAttack	T-PGD	SemAttack
Fine-Tuned	8.8	41.5	3.0	1.3
InfoBERT	32.1	47.1	25.4	2.0
FreeLB++	19.6	58.2	15.5	3.4
RSMI	79.6	n/a	43.3	n/a
DiffuseDef-FreeLB++	59.4	68.1	50.3	28.2
DiffuseDef-RSMI	81.7	n/a	48.4	n/a

Table 5.5: Accuracy under white-box attacks.

5.5.3 Ablation Analysis

To understand how each component contributes to DiffuseDef, we conduct an ablation analysis on the AGNews dataset by disabling ensembling, diffusion denoising, and adversarial training one at a time. Table 5.6 shows the ablation results, and we find that removing either component (i.e., ensembling and diffusion denoising) at inference time leads to a sharp decrease in accuracy under attack, which indicates that both denoising and ensembling contribute significantly to the improved robustness. Nevertheless, removing adversarial training, though also introducing a degraded AUA score, is less detrimental than removing the other inference time components, showing that DiffuseDef is not dependent on a specific adversarial training method.

Method	TF	TB	BA
DiffuseDef	84.5	86.0	84.6
w/o ensembling	64.2	65.8	54.9
w/o diffusion denoising	57.1	58.4	47.7
w/o adv training	80.3	80.9	80.5

Table 5.6: Ablation results for DiffuseDef’s accuracy under attack on AGNews datasets.

5.5.4 DiffuseDef + Adversarial Augmentation

In Figure 5.2, we combine DiffuseDef with an adversarial data augmentation method and evaluate the robustness on AGNews. As a replacement for adversarial training, adversarial augmentation trains on synthetic training data using text adversarial examples generated by the attacker. We first generate adversarial augmentation data using TextFooler to attack the

fine-tuned BERT model. We then train the classifier with the original training set plus the successful attack data. Subsequently, the augmented classifier is further performed with diffusion training following the same method, resulting in a DiffuseDef variant. We notice that adversarial training on augmented data helps improve the performance on all three attacks. After applying diffusion training, the adversarial performance significantly improves over the augmented model, which proves the flexibility of DiffuseDef to be applied with different types of adversarial training methods.

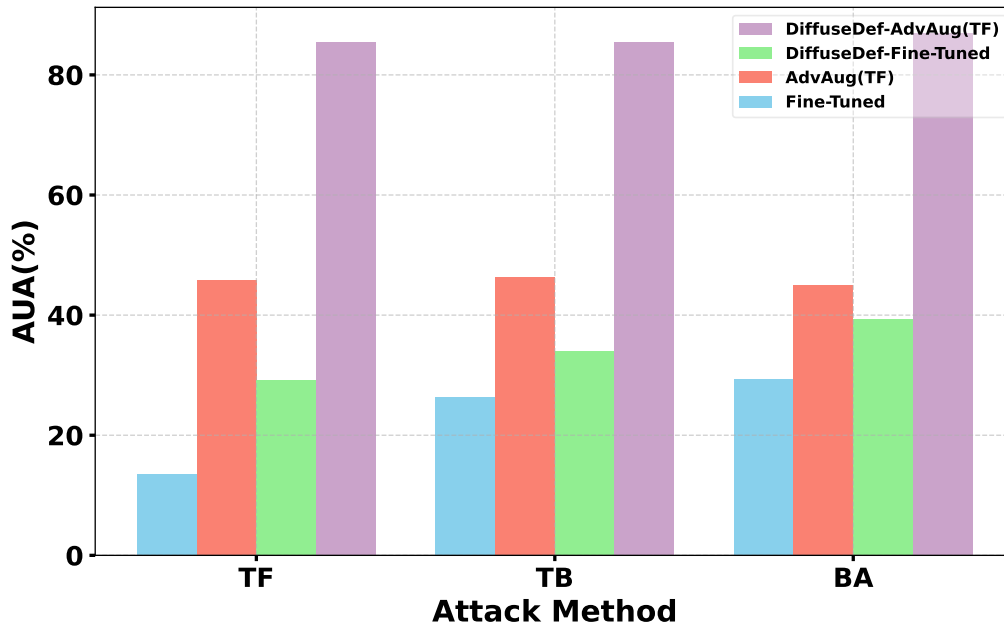


Figure 5.2: Robustness of DiffuseDef with textual adversarial augmentation method.

5.5.5 Robustness w.r.t Token Length

Figure 5.3 provides a comparison of the defense rate for different models by token length on the IMDB dataset. The defense rate is calculated as the percentage of test examples in which TextFooler fails to construct a successful attack. All models except RSMI show a consistent trend that the defense rate declines as the texts lengthen. This trend can be attributed to the nature of adversarial attacks, as longer texts allow for the generation of more adversarial examples. Specifically, adversarial training defense methods like InfoBERT and FreeLB++ show poor performance on longer texts (more than 300 tokens), with the defense rate reduced to nearly 0. This drastic decline indicates that, given an adequate number of queries, the

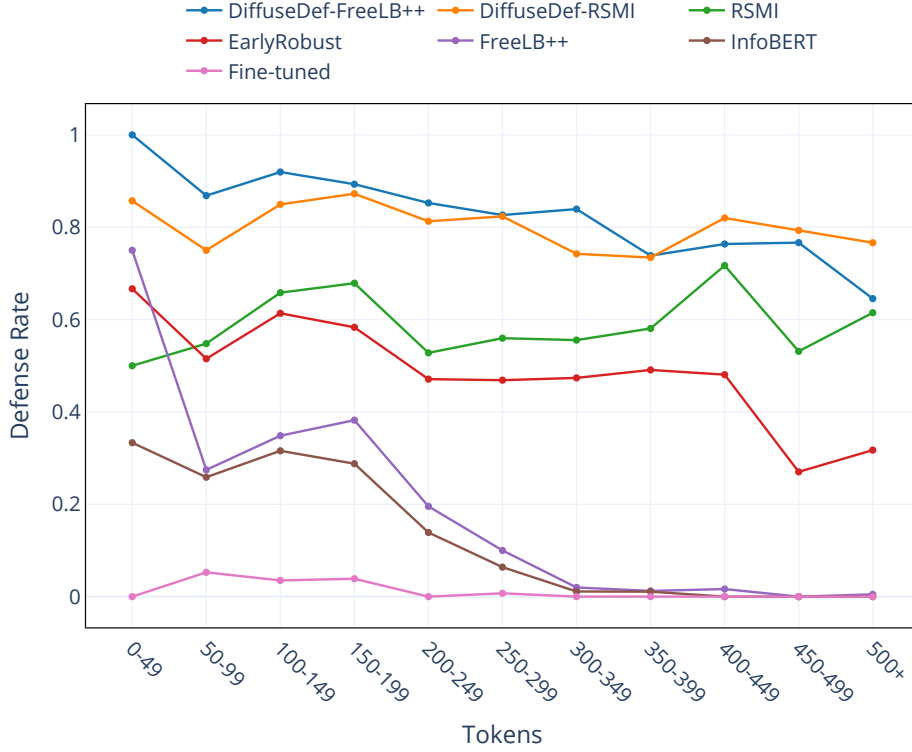


Figure 5.3: Defense rate (on TextFooler) w.r.t token length for different models on IMDB dataset.

attacker is guaranteed to find a successful attack to fool these models. Similarly, EarlyRobust exhibits a performance drop on long texts as it is based on FreeLB training. RSMI, however, performs worse on short texts, but its defense rate increases as the text length grows. Compared to all SOTA defense approaches, the two DiffuseDef variants show a more steadily declining trend and maintain a higher defense rate across all token lengths, i.e., DiffuseDef is more robust to input text length.

5.5.6 Effect of Additional Denoising Steps

In Figure 5.4, we study how the inference denoising steps t' can affect the adversarial performance. For the DiffuseDef model without ensembling, both the AUA score and the number of queries required to attack increase as the inference denoising step is larger. As the denoising step t' grows from 1 to 30, the AUA score improves from 58 to 65 while the number of attack queries grows from 430 to 780. In contrast, for DiffuseDef with ensembling, the model maintains a stable but robust performance in AUA and number of queries, regardless of the increase of

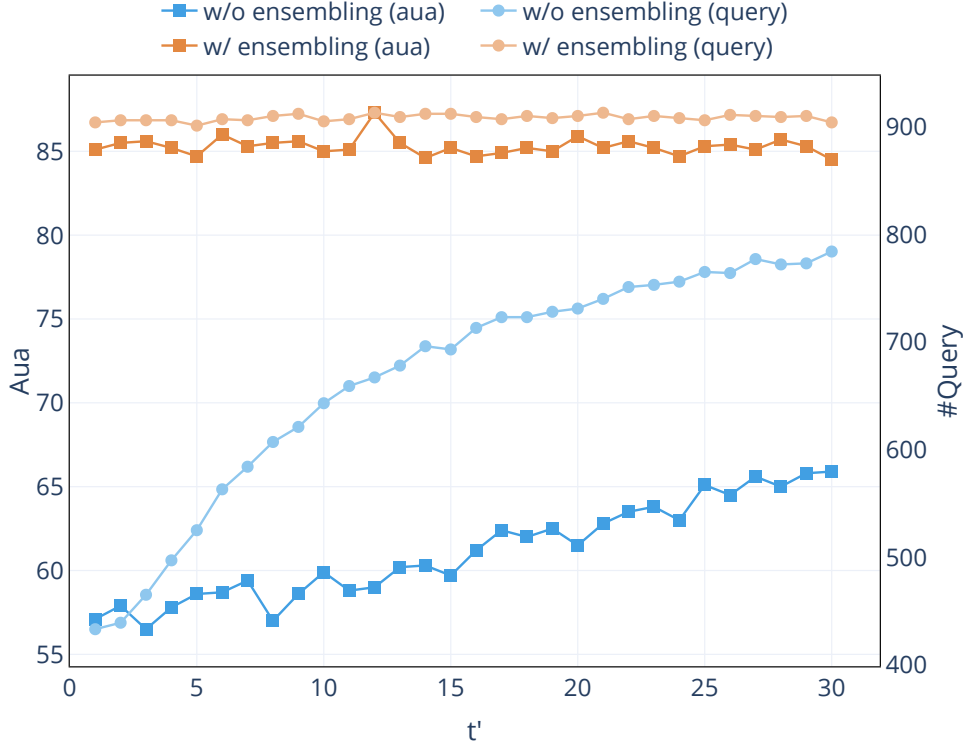


Figure 5.4: AUA and #Query (TextFooler) w.r.t inference denoising step for DiffuseDef w/ and w/o ensembling.

t' . Considering that the ensembling introduces a notable performance increase, the DiffuseDef model is likely to be hitting an upper bound in both metrics, thus no further improvement is reached by increasing the denoising steps. However, it also shows that with ensembling, DiffuseDef can be applied with a smaller t' for better efficiency while maintaining a robust adversarial performance.

5.5.7 Ensembling Diffused Hidden Representations

In DiffuseDef, the text hidden state is diffused and ensembled to form a denoised hidden representation, which contributes significantly to the improved adversarial robustness. In this section, we study how the ensembling of diffused hidden representations helps defend against adversarial attacks.

As mentioned in Section 2.2.2.1, attack methods need to first rank token importance based on its influence on prediction. Specifically, the importance score is calculated by comparing the

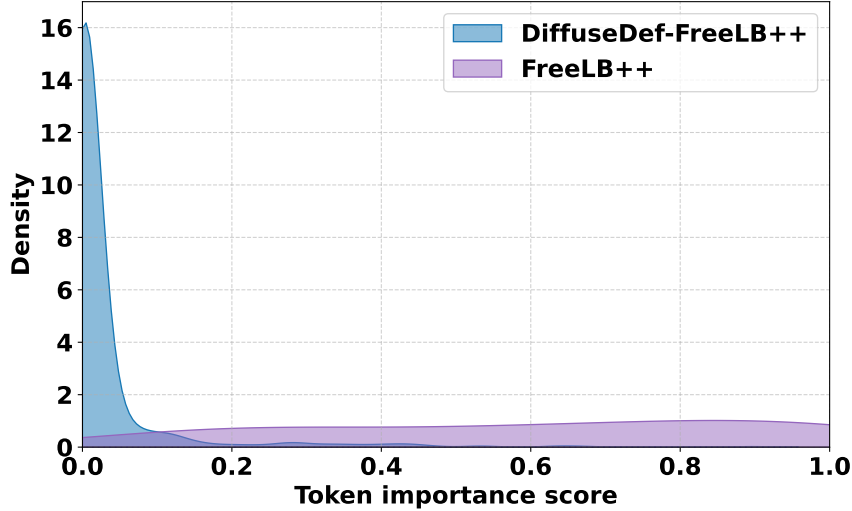


Figure 5.5: Distribution of max token importance score in the AGNews test set.

changes in model prediction probabilities after removing each word. In Figure 5.5, we compare the density distribution of the max token importance score between FreeLB++ and its DiffuseDef counterpart. DiffuseDef shows a notably lower percentage than FreeLB++ when the max importance score is high, where the attacker can easily find the vulnerable token to construct adversarial examples. This difference shows that DiffuseDef can complicate the process of important word searching, which accounts for the increased number of queries required for a successful attack.

Method	L2	Cosine
FreeLB++	12.53	0.35
DiffuseDef-FreeLB++	10.66	0.27
RSMI	9.72	0.24
DiffuseDef-RSMI	8.61	0.21

Table 5.7: L2 and cosine distance between hidden states for clean and adversarial texts.

In addition, DiffuseDef mitigates the difference between clean and adversarial texts by reducing the distance between their hidden states. In Table 5.7, we report the L2 and cosine distance between clean and adversarial hidden states for FreeLB++ and RSMI. Both show lower L2 and cosine distance after applying DiffuseDef, indicating that ensembling diffused representation repositions the adversarial example closer to the clean example, leading to the model maintaining its predictions.

5.5.8 Efficiency of DiffuseDef

Given that DiffuseDef adds additional denoising and ensembling steps during inference, it inevitably increases the computation time compared to its base model. To study its efficiency, we report the number of model parameters, inference FLOPS, and training time in Table 5.8. In addition to the defense methods in Table 5.4, we also compare the efficiency of DiffuseDef with two other SOTA ensembling-based defense methods, i.e., RanMask (Zeng et al., 2023) and SAFER (Ye et al., 2020).

Method	Params	FLOPS	Train
Fine-Tuned (BERT)	110M	46G	1x
EarlyRobust	82M	32G	0.8x
FreeLB++	110M	46G	10.5x
InfoBERT	110M	46G	6.5x
RSMI	110M	92G	1.25x
RanMask ($k = 10$)	110M	459G	1.2x
SAFER ($k = 10$)	110M	459G	1x
DiffuseDef ($t' = 1, k = 10$)	120M	96G	1.1x
DiffuseDef ($t' = 5, k = 10$)	120M	267G	1.1x

Table 5.8: Efficiency comparison of DiffuseDef and other methods. Params: number of model parameters. FLOPS: number of floating point operations per second at inference time, calculated with batch size of 1 and sequence length of 256. Train: training time.

All SOTA models have the same number of parameters as the fine-tuned BERT model, except EarlyRobust, which applies attention head pruning for better efficiency. DiffuseDef requires more inference FLOPS than non-ensembling-based baselines such as FreeLB++ and EarlyRobust, due to the diffusion steps. With $t' = 1$ and $k = 10$, the FLOPS for DiffuseDef doubles from 46G to 96G, nevertheless, this number is close to the RSMI model as it requires gradient information during inference. Despite this increase, DiffuseDef is more efficient than ensemble-based methods like RanMask and SAFER, which require a full forward pass for all ensembles. With the same ensembling number of 10, both RanMask and SAFER require 459G FLOPS, which is 10x the number for the BERT baseline. In contrast, even with t' increased to 5, DiffuseDef can be run faster with 267G FLOPS, mitigating the efficiency problem from ensembling while maintaining the benefit of improved robustness.

Regarding training time, although training the diffusion layer requires more epochs, it costs a similar time as training the classifier due to the small size of the diffusion layer. By contrast, adversarial training methods such as FreeLB++ and InfoBERT require much longer training time due to increased iterations of back-propagation per training batch.

5.5.9 Example of Noising and Denoising in DiffuseDef

t'	MLM Output (add noise)	MLM Output (add noise then denoise)
0	IBM Chips May Someday Heal Themselves New technology applies electrical fuses to help identify and repair faults.	-
5	the ibm chips may someday heal themselves new technology introduces electrical fuses to help identify and repair faults.	the ibm chips may someday heal themselves new technology introduces electrical fuses to help identify and repair faults.
6	) ibm chips may someday heal themselves new technology introduces electrical fuses to help identify and repair faults.	the ibm chips may someday heal themselves new technology introduces electrical fuses to help identify and repair faults.
7	the. chips may someday heal themselves new technology introduces electrical fuses to help identify and repair faults.	the. chips may someday heal themselves new technology uses electrical fuses to help identify and repair faults.
8	).. may someday heal themselves new technology introduces electrical fuses to help identify and repair faults.	the.ibm. may someday heal themselves new technology uses electrical fuses to help identify and repair faults.
9	the ibm chips may someday heal themselves new technology uses electrical fuses to help identify and repair faults.	the ibm chips may someday heal themselves new technology introduces electrical fuses to help identify and repair faults.
10	the. chips may someday heal themselves new technology extends electrical fuses to help identify and repair faults.	the.ibm. may someday heal themselves new technology develops electrical fuses to help identify and repair faults.

Table 5.9: MLM outputs from hidden states with noise added and hidden states with first noise added but then denoised. We only report t' above 5 as the MLM outputs with smaller t' are identical to the clean text.

Adding and removing noise to hidden states are essential features in DiffuseDef, which contribute to the improved adversarial robustness. To study how adding or removing noise affects the semantic meaning of the text, we feed the hidden states to a not fine-tuned BERT model with the masked language modeling (MLM) head to generate the text output.

In Table 5.9, we present the MLM outputs from hidden states added with different steps of noise and the MLM outputs from noise hidden states denoised with the same number of steps. In the example shown, with more noise added, some semantic information can be lost and replaced by symbols or function words like "." or "the". In contrast, denoising for the same

number of steps helps alleviate such information loss. For example, the word "IBM" can be recovered from the noise.

t'	Clean Text / MLM Output	Adv Text / MLM Output	Pred clean	Pred adv
0	United Apology over Website Abuse Manchester United have been forced to issue an embarrassing apology to Liverpool for an ill-advised attack on the Anfield outfit on its own website.	United Apology over Website Abuse Manchester Nations have been forced to issue an embarrassing apology to Liverpool for an ill-advised attack on the Anfield outfit on its own website.	Sports	World
1	football. apology over website abuse manchester united have been to issue an embarrassing apology to liverpool for an the advised attack on the anfield outfit on its own website.	the. apology over website abuse manchester nations have been the to issue an embarrassing apology to liverpool for an the advised attack on the anfield outfit on its own website.	Sports	World
2	the. apology over website abuse manchester united have been to issue an embarrassing apology to liverpool for an the advised attack on the anfield outfit on its own website.	the. apology over website abuse manchester nations have been the to issue an embarrassing apology to liverpool for an the advised attack on the anfield outfit on its own website.	Sports	World
3	the. apology over website abuse manchester united have been the to issue an embarrassing apology to liverpool for an the advised attack on the anfield outfit on its own website.	the. apology over website abuse manchester s have been the to issue an embarrassing apology to liverpool for an the advised attack on the anfield outfit on its own website.	Sports	Sports
4	the. apology over website abuse manchester united have the to issue an embarrassing apology to liverpool for an the advised attack on the anfield outfit on its own website.	the. apology over website abuse manchester s have been the to issue an embarrassing apology to liverpool for an the advised attack on the anfield outfit on its own website.	Sports	Sports
5	the. apology over website abuse manchester united have the the to issue an a apology to liverpool for an the advised attack on the anfield outfit on its own website.	the. apology over website abuse manchester united have been the to issue an the apology to liverpool for an advised attack on the anfield outfit on its own website.	Sports	Sports

Table 5.10: MLM outputs and model predictions from ensembling diffused hidden states at different denoising steps.

However, in practice, it is not realistic to assume the number of denoising steps. Therefore, in Table 5.10 we show the MLM outputs of denoised hidden states directly from clean and adversarial texts. On clean text, we observe that a higher number of denoising steps can result in more abstraction of the texts. For example, more words are replaced with "the" in the MLM outputs as t' grows. However, words related to the topic (e.g., "Manchester United", "Liverpool") are kept during the denoising process, thus the model can predict the target label (Sports) correctly. Similarly, the trend of abstraction can also be found on adversarial text while

we observe that the denoising can help remove the adversarial noise/perturbation and recover the word "united" from "nation", thus resulting in its correct prediction on the adversarial text.

5.6 Conclusions

In this chapter, we focus on improving robustness against textual adversarial attacks and propose a novel adversarial defense method, DiffuseDef. It combines adversarial training, diffusion training, and ensembling to improve model robustness to black-box adversarial attacks. DiffuseDef can build on any existing adversarial training method, training an additional diffusion layer to predict and remove randomly sampled noise at a given timestep. During inference, the diffusion layer is used to denoise the adversarial hidden states, which are ensembled to construct a robust text representation. Our experiments validate the effectiveness and efficiency of DiffuseDef, which significantly outperforms SOTA on common black-box and white-box adversarial attack methods, and maintains the performance on clean texts. Analysis shows that DiffuseDef makes it difficult to find vulnerable tokens to attack, and also reduces the difference between the hidden representations of clean and adversarial texts.

Reflecting on the research questions, this chapter answers research question 2 by proposing a diffusion-based denoising approach, which significantly improves the robustness to adversarial attacks. This method is also implemented in a plug-and-play manner and can be applied to any model and combined with any adversarial training method. Results also validate the effectiveness of its generalization to different adversarial attacks.

Chapter 6

Adversarial Multi-task Training for Debiased Toxicity Detection

© [2023], Epic Games, Inc.

Warning: *This chapter contains examples that may be toxic or offensive.*

In the chapters 3, 4, and 5, we mainly focused on tackling induced vulnerabilities, including textual noise and adversarial attacks. Note that vulnerabilities, noise and adversarial attacks, all exploit weaknesses in models by perturbing inputs at inference time, leading to a significant decline in performance compared to clean, unattacked inputs. Therefore, improving robustness to these induced vulnerabilities is equivalent to bridging the gap between standard (clean) inputs and non-standard (noisy/adversarial) inputs. For example, the usage of visual and conversational contexts can compensate for the information loss from noisy texts, and the diffusion-based denoising maps the noisy, adversarial hidden states back to the clean hidden states.

However, the model’s robustness can be upper-bounded by the model’s performance on the original clean data, where learned vulnerabilities can also be obtained during the training stage. For example, one of the common types of learned vulnerability is unintended biases, such as gender bias, identity bias, etc, and such vulnerabilities can be learned from the data

distribution due to spurious correlations in the data. Addressing learned vulnerabilities can be a more challenging task because they may set the upper limit on any robustness we can achieve at inference time. In this chapter, we turn our attention to toxicity detection, where the model is likely to learn unintended biases by correlating toxicity with other relevant tasks like profanity or negative semantics, aiming to improve the model’s robustness to bias.

Current toxicity detection models tend to overfit specific terms such as profanity words or group identity mentions, among others. This results in unintended biases in predictions, such as models predicting all occurrences of “immigrants” as toxic. Previous work on mitigating biases in toxicity detection focuses on improving models via techniques such as regularization, data sampling, and ensembling. In this chapter, we propose a new method where potential biases are modeled as additional tasks. Our multi-task learning approach extracts representations that share relevant information across tasks, and relies on adversarial learning to disentangle representations that are task-specific and could lead to biases. Our experiments on three potential biases – sentiment, profanity, and identity mentions – show that our method successfully mitigates unintended biases and improves toxicity detection performance on out-of-domain settings while also improving or maintaining performance on in-domain settings.

The work presented in this chapter was conducted when Zhenhao worked in Epic Games and has been published as a patent “Multi-task neural network for toxicity detection” [US20230419117A1] (Specia et al., 2023) by **Lucia Specia, Marina Fomicheva, Ozan Caglayan, Zhenhao Li**. By the time of this thesis submission, all authors of the patent are employees of **Epic Games**, and as such, the copyright of this chapter is held by Epic Games.

6.1 Introduction

Toxicity detection, also referred to as abusive-language, hateful-speech, or offensive-content detection, aims to automatically flag text containing attacks, insults, harassment, threats, hate speech, or other language likely to harm a specific person or a group (Schmidt and Wiegand, 2017; Fortuna and Nunes, 2018b). It has been a crucial step in content-moderation pipelines

for social media, forums, and online games, where manual review alone cannot keep pace with the volume of growing user-generated content. In recent years, toxicity detection has been an important topic in the natural language processing community (Qian et al., 2018; Zhang et al., 2018a; Vidgen et al., 2021; Hartvigsen et al., 2022). The wide adoption of Transformer-based (Vaswani et al., 2017) pre-trained language models such as BERT (Devlin et al., 2019), RoBERTa (Liu et al., 2019c), etc., led to substantial improvements in the performance of toxicity detection models (Mathew et al., 2021). However, current pre-trained language models tend to overfit on specific terms such as profanity words, group identity mentions, negative sentiment phrases, etc. Thus, unintended biases might be introduced when predicting toxicity (Nozza et al., 2019; Nangia et al., 2020; Vig et al., 2020; Ousidhoum et al., 2021; Bauer et al., 2021; Röttger et al., 2021; Zhou et al., 2021a; de Vassimon Manela et al., 2021).

Table 6.1 lists some examples of potential biases in current toxicity detection models. For instance, models may predict texts that include the profanity word “Damn” as toxic, but in reality, this word can also be used as an intensifier for positive emotions. Likewise, mentions of certain identity groups and negative sentiment words tend to trigger models to misclassify cases as toxic. With sentiment words, the opposite is also true: positive words like *love* can lead the model to classify toxic cases as non-toxic incorrectly. Previous work on mitigating biases in toxicity detection do so via techniques such as regularization (Kennedy et al., 2020a; Kaneko and Bollegala, 2021), data sampling (Dixon et al., 2018; Park et al., 2018), and ensembling (Halevy et al., 2021). However, these techniques are limited in that they either require prior knowledge (e.g. a set of biased identity terms) or data augmentation on the bias terms, which does not generalize to different terms or types of biases. Recent work use gradients (Bose et al., 2022) or attention (Attanasio et al., 2022) information to debias the model with regularization loss, which avoids the need for pre-defined bias terms, but they focus on single-task training and do not consider related tasks.

In this work, we propose a new method to debias toxicity detection by modeling potential biases as additional tasks. We employ a multi-task learning framework, which extracts feature representations that share relevant information across tasks, benefiting toxicity detection by learning from other related tasks. Crucially, we propose an adversarial learning technique to

Bias	Toxic	Non-toxic
Profanity	<i>Damn! I hate you!</i>	<i>Damn! I love you!</i>
Identity	<i>Fat people won't fit.</i>	<i>Fat people won't fit narrow beauty standards.</i>
Sentiment	<i>I hate these guys!</i>	<i>I hate these cars!</i>

Table 6.1: Examples of potential biases in toxicity detection: current models tend to overfit on the bias term and predict toxicity based on its occurrence.

disentangle task-specific representations to mitigate biases.

We perform experiments on three potential biases, namely sentiment, profanity, and identity mentions. Experimental results show that (i) our method can mitigate biases and improve out-of-domain toxicity detection performance and (ii), while doing so, it also maintains or improves the performance on in-domain settings.

In the following sections, we first introduce the related work (§ 6.2) around our proposed adversarial multi-task training method. Then we describe the proposed method and model architecture (§ 6.3), followed by the detailed experimental settings (§ 6.4). The experimental results on sentiment, profanity and identity mention bias tasks are presented in § 6.5, § 6.6, and § 6.7, respectively. Finally, we conclude this work and discuss the future directions.

6.2 Related Work

6.2.1 Toxicity Detection with MTL

Toxicity prediction is a complex problem that relates to many other tasks. Recent work has addressed this problem from a multi-task learning perspective. For example, Vaidya et al. (2020) propose modeling toxicity with identity mention tasks and use pseudo-labels for data augmentation. Ousidhoum et al. (2019) propose the MLMA dataset and train multi-task models on toxicity, sentiment, target attribute, target group, and directness tasks. Plaza-del Arco et al. (2021) propose to learn toxicity jointly with sentiment, emotion recognition, and identity mentions tasks. Finally, Zhou et al. (2021b) propose a multi-task model that jointly models

toxicity and sentiment tasks through feature sharing. Although these works show improvements in general toxicity detection with the help of the shared knowledge from other tasks, no attempt has been made to remove unwanted biases that emerge when models make toxicity predictions based on the spurious correlations introduced by the other tasks.

6.2.2 Disentangling Representations

One possible approach for debiasing in MTL models is to learn disentangled representations for each task. Therefore, the MTL model can benefit from both shared and task-specific representations. This approach has been applied in domain adaptation and multi-domain modeling in computer vision (Ganin and Lempitsky, 2015; Tzeng et al., 2015; Bousmalis et al., 2016; Yang et al., 2019a; Wang et al., 2020) and other tasks in natural language processing (Liu et al., 2017; Peng et al., 2018; Chen and Cardie, 2018; Cai and Wan, 2019; Wu and Guo, 2020; Du et al., 2020; Wu et al., 2021).

Ganin and Lempitsky (2015) first propose the gradient reversal layer (GRL) and use a domain classifier to help learn a domain-invariant feature representation. Bousmalis et al. (2016) propose the domain separation network, which learns shared and private features and utilizes orthogonality constraints and a domain classifier to ensure the private features are separated from each other while the shared features are common. Yang et al. (2019a) propose TACO, which consists of a shared encoder, a separate encoder for each domain, and a domain discriminator. They optimize the domain-specific feature with adversarial loss so that the specific feature only performs well on its own domain. Wang et al. (2020) benchmark various adversarial training approaches to learn domain-invariant features.

Domain-adversarial training is also commonly applied in text classification. Liu et al. (2017) adopt the domain separation network architecture and replace the encoder with LSTM (Hochreiter and Schmidhuber, 1997b) for text classification tasks. Chen and Cardie (2018) propose Multinomial Adversarial Network to learn domain-invariant features for multi-domain text classification. Cai and Wan (2019) further extend the model architecture by utilizing domain

features when classifying the text. Recently, models with shared and private features are prevalent in multi-domain classification, such as DACL (Wu and Guo, 2020), and CAN (Wu et al., 2021). The shared feature is optimized adversarially with a domain discriminator to ensure its domain-invariance, while the private features are optimized with a regularization loss to make them different from each other.

Our proposed architecture is inspired by the TACO model (Yang et al., 2019a), but extends it from CNN to transformer models. Moreover, our approach differs from theirs in that we focus on the multi-task setting rather than the multi-domain setting. For the multi-domain setting, inputs from different domains are different (e.g. with different writing styles, discussing different topics, etc.), whereas in our case, the inputs for different tasks are the same. Therefore, our method extracts different task-specific features from the same input, making it more challenging than previous work.

6.3 Method

In this section, we introduce our adversarial multi-task model (shortened to MTL-adv) with shared (also referred to as “task-common”) and private (also referred to as “task-specific”) task features. By sharing common knowledge across tasks through the task-common features, we expect to increase the robustness of the model. By introducing task-specific features, i.e. by disentangling toxicity-specific representations from other related tasks, we expect to mitigate the potential biases that these tasks might introduce.

Figure 6.1 showcases an example realization of the proposed model with two tasks. The model consists of a pre-trained language model (PLM) $\mathcal{P}$, a task-common feature extractor $\mathcal{F}_c$, a task-specific feature extractor $\mathcal{F}_s^{(i)}$ for each task i , a task classifier $\mathcal{C}^{(i)}$ for each task i , and a task discriminator $\mathcal{D}$. The feature extractors, task classifiers, and task discriminators are all MLP-based networks.

Given an input text T and the task labels $y^{1..N}$, the pre-trained language model $\mathcal{P}$ first generates a vector $v = \mathcal{P}(T)$ as the sentence representation. In our case, $\mathcal{P}$ is implemented by

BERT (Devlin et al., 2019), and the vector representation is the hidden state of the [CLS] token.

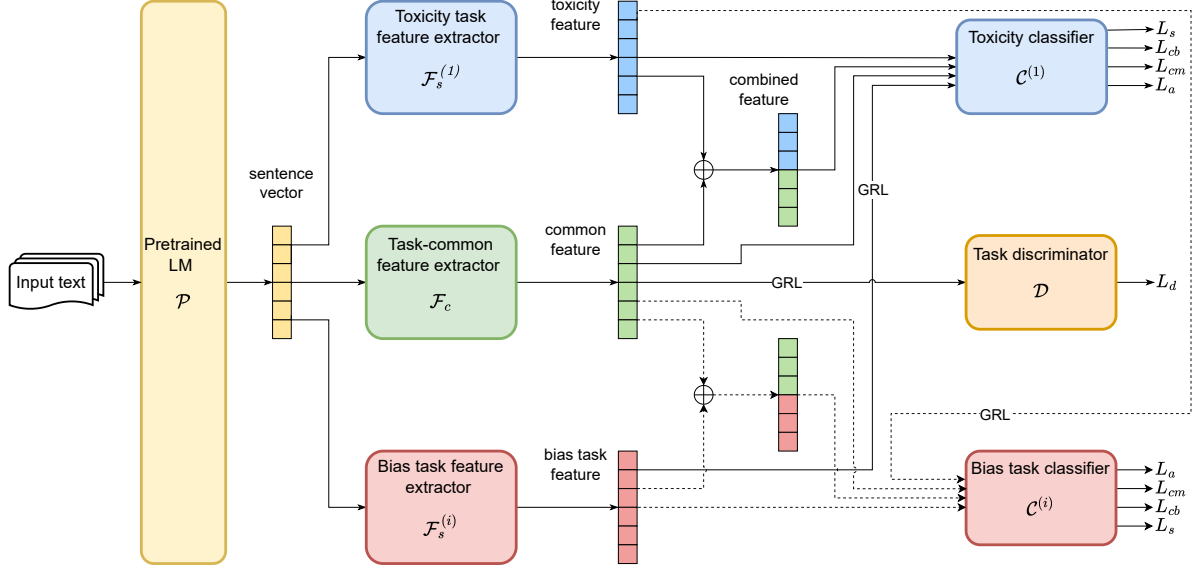


Figure 6.1: An example of our adversarial multi-task model with two tasks. Solid lines represent loss computation for the toxicity task, while dotted lines represent the loss computation for the bias task.

The task-common feature extractor $\mathcal{F}_c$ transforms the sentence vector v into the task-common feature f_{cm} , which is shared across all tasks. The task-specific feature extractor $\mathcal{F}_s^{(i)}$ for task i transforms v into the task-specific feature $f_s^{(i)}$, which is **only** used for the corresponding task i . The two types of features are concatenated to form the combined feature f_{cb} , which embodies both shared and private information:

$$\begin{aligned}
 f_{cm} &= \mathcal{F}_c(v) \\
 f_s^{(i)} &= \mathcal{F}_s^{(i)}(v) \\
 f_{cb}^{(i)} &= [f_{cm}; f_s^{(i)}] \\
 \forall i &\in \{1, 2, \dots, N\}
 \end{aligned} \tag{6.1}$$

The three features are then fed to task classifiers $\mathcal{C}^{(i)}$ to calculate the common loss $\mathcal{L}_{cm}$, specific

loss $\mathcal{L}_s$, and the combined loss $\mathcal{L}_{cb}$ for each task i :

$$\begin{aligned} L_{cm} &= \sum_{i=1}^N \ell^{(i)}(\mathcal{C}^{(i)}(f_{cm}), y^{(i)}) \\ L_s &= \sum_{i=1}^N \ell^{(i)}(\mathcal{C}^{(i)}(f_s^{(i)}), y^{(i)}) \\ L_{cb} &= \sum_{i=1}^N \ell^{(i)}(\mathcal{C}^{(i)}(f_{cb}^{(i)}), y^{(i)}) \end{aligned} \quad (6.2)$$

where $\ell^{(i)}$ corresponds to the loss function for task i , e.g. BCE loss for binary classification and CE loss for multi-class classification. By having a dedicated loss for each feature type, the model is optimized so that all feature representations perform well on their corresponding tasks.

Next, we want to differentiate the task-common and the task-specific features. Since the task-common feature extractor is optimized for all tasks, it may learn specific patterns when optimizing for different tasks. To make the task-common feature task-invariant, we use a task discriminator $\mathcal{D}$ to distinguish which task the feature is from. The discriminator minimizes the discrimination loss $\mathcal{L}_d$ while the task-common feature extractor attempts to maximize it. Although the input text is the same for all tasks, we change the expected label, which is the task ID, when optimizing the discrimination loss. This loss acts as a regularization so that the task-common feature extractor produces a task-invariant representation. After training, the discriminator is no longer used.

$$L_d = \sum_{i=1}^N \text{CE}(\mathcal{D}(f_{cm}), i) \quad (6.3)$$

To achieve this MINMAX optimization, we apply a Gradient Reversal Layer (Ganin and Lempitsky, 2015) to f_{cm} and feed it to the task discriminator. A GRL is equivalent to an identity transformation during the forward pass, but multiplies the gradient by a negative scalar during back-propagation.

Regarding the task-specific features, we expect them to be beneficial **only** for a particular task,

but not useful for other tasks. When optimizing the task-specific feature $f_s^{(i)}$ for the task i , we obtain task-specific features for all other tasks $f_s^{(j)}$ where $j \in [1, 2, \dots, N], j \neq i$ from $\mathcal{F}_s^{(j)}$. These task-specific features are passed through GRL and then fed to the classifier $\mathcal{C}^{(i)}$ for task i . The adversarial loss $\mathcal{L}_a$ is then computed and is maximized when there is a mismatch between the task-specific feature and the task classifier:

$$L_a = \sum_{i=1}^N \sum_{j \neq i} \ell^{(i)}(\mathcal{C}^{(i)}(f_s^{(j)}), y^{(i)}) \quad (6.4)$$

The final MTL objective can now be formulated as:

$$\underset{\{\mathcal{P}, \mathcal{F}_c, \mathcal{F}_s^i, \mathcal{C}^i\}}{\text{MIN}} \underset{\{\mathcal{D}, \mathcal{F}_s^j\}}{\text{MAX}} \mathcal{L}_{cm} + \mathcal{L}_s + \mathcal{L}_{cb} - \alpha_1 * \mathcal{L}_d - \alpha_2 * \mathcal{L}_a \quad (6.5)$$

where α_1, α_2 are coefficients for the discrimination and adversarial losses, respectively. Since we focus on the shared and private aspects of f_{cm} and f_s , which mainly depend on the feature extraction layers, we freeze the pre-trained language model component when optimizing the discrimination and adversarial losses. At inference time, only the combined feature f_{cb} is used to make a prediction. The training procedure of our proposed adversarial multi-task model is shown in Algorithm 2.

6.4 Experiments

6.4.1 Data

We use three datasets with three different bias tasks to debias the model for toxicity detection. Each dataset contains a binary toxicity label and a bias task label for the every data sample. All dataset statistics are presented in Table 6.2, grouped according to different datasets.

First, we use the Measuring Hate Speech (MHS) dataset¹ (Kennedy et al., 2020b) to debias toxicity with a **sentiment** classification task. We randomly sample two sets of 1,000 examples

¹CC-BY-4.0 license. <https://huggingface.co/datasets/ucberkeley-dlab/measuring-hate-speech>

Algorithm 2: Adversarial MTL training

Input: input text T , task labels $y^{1..N}$
Output: $\mathcal{L}_s, \mathcal{L}_{cm}, \mathcal{L}_{cb}, \mathcal{L}_a, \mathcal{L}_d$

```

1  $v \leftarrow \mathcal{P}(T)$ ;
2  $f_{cm} \leftarrow \mathcal{F}_c(v)$ ;
3 for  $i \leftarrow 1$  to  $N$  do
4    $y'_{cm} \leftarrow \mathcal{C}^{(i)}(f_{cm})$ ;
5   Compute  $\mathcal{L}_{cm}^{(i)}(y'_{cm}, y^{(i)})$ ;
6    $f_s^{(i)} \leftarrow \mathcal{F}_s^{(i)}(v)$ ;
7    $y'_s \leftarrow \mathcal{C}^{(i)}(f_s)$ ;
8   Compute  $\mathcal{L}_s^{(i)}(y'_s, y^{(i)})$ ;
9    $f_{cb}^{(i)} \leftarrow [f_{cm}^{(i)}; f_s^{(i)}]$ ;
10   $y'_{cb} \leftarrow \mathcal{C}^{(i)}(f_{cb})$ ;
11  Compute  $\mathcal{L}_{cb}^{(i)}(y'_{cb}, y^{(i)})$ ;
12  Apply Gradient Reversal to  $f_{cm}^{(i)}$ ;
13   $y'_{task} \leftarrow \mathcal{D}(f_{cm})$ ;
14  Compute  $\mathcal{L}_d^{(i)}(y'_{task}, i)$ ;
15  for  $j \leftarrow 1$  to  $N, j \neq i$  do
16     $f_a^{(j)} \leftarrow \mathcal{F}_s^{(j)}(v)$ ;
17    Apply Gradient Reversal to  $f_a^{(j)}$ ;
18     $y'_a \leftarrow \mathcal{D}(f_a^{(j)})$ ;
19    Compute  $\mathcal{L}_a^{(i,j)}(y'_a, y^{(i)})$ ;
20  end
21 end
22 Sum  $\mathcal{L}_s, \mathcal{L}_{cm}, \mathcal{L}_{cb}, \mathcal{L}_a, \mathcal{L}_d$  over  $i$  and  $j$ ;

```

to form the dev and test sets.

Next, we select **profanity** word detection as the second bias task. We use the Hateful Founta (HF) dataset² (Founta et al., 2018) as it contains a large proportion of samples with profanity words. We use a high-precision profanity lexicon to define the set of profanity words. We remove samples with the “spam” label and focus on binary toxicity labels. We randomly sample 2000 examples from the dataset to form each of the dev and test sets. In addition, we also use the Implicit Hate dataset³ (ElSherief et al., 2021) as an out-of-domain (OOD) test set, since it contains adversarial cases: implicitly toxic samples without profanity words, as well as non-toxic samples with profanity words. A debiased model is expected to perform better on this test set as the toxicity and profanity are less correlated.

²CC-BY-4.0 license. https://zenodo.org/records/3678559#.Xk_23y97FhE

³MIT license. <https://github.com/SALT-NLP/implicit-hate>

	TRAIN		DEV		TEST	
	NON	TOX	NON	TOX	NON	TOX
MHS Dataset						
Negative	11,796	13,127	339	348	303	392
Neutral	7,394	97	187	0	180	0
Positive	4,089	6	126	0	125	0
Hateful Founta (HF) Dataset						
NOT	44,307	1,916	1,183	43	1,199	50
PROF	5,980	21,630	163	611	152	599
Implicit Hate OOD Test Set						
NOT	-	-	-	-	10,299	5,751
PROF	-	-	-	-	2,978	2,421
Unintended Bias (UB) Dataset						
NOT	1.22M	98.9K	2,207	193	130K	10.7K
IDEN	94.1K	21.3K	1,404	325	9.9K	2,257

Table 6.2: Statistics of toxicity labels broken down into sentiment, profanity and identity debiasing tasks, across four datasets. TOX: toxic. NON: non-toxic. NOT: no profanity/identity. PROF: has profanity. IDEN: has identity.

Finally, we use the **identity** mention detection task as the third bias task. The aim of this task is to detect whether the text contains any identity mentions, such as race, gender, ethnicity, etc. We use the Jigsaw Unintended Bias in Toxicity Classification (UB) dataset⁴, which contains 1.8M examples annotated with toxicity labels, 0.4M of which are annotated with identity labels. We use the official test split as our test set and randomly sample 4000 examples with identity annotations as our dev set. All 24 identity types are used for the bias task, making this a multi-label classification task.

All datasets are pre-processed with the same pipeline: hashtags, URLs, emails, and usernames are removed; emojis are transformed into corresponding phrasal forms; texts are lower-cased and filtered to have a minimum and maximum of 5 and 100 words, respectively.

⁴CC license. <https://www.kaggle.com/c/jigsaw-unintended-bias-in-toxicity-classification>

6.4.2 Training and Evaluation

We use a pre-trained BERT model (Devlin et al., 2019) (`bert-base-uncased`) as a starting point for each dataset. We compare our models with two baselines: a single-task BERT model (STL) fine-tuned only on the toxicity task, and a vanilla multi-task BERT model (MTL) fine-tuned with two classifier heads: one for toxicity and one for the bias task. We perform hyperparameter tuning for all models on the three datasets and select the best-performing hyperparameter settings on the dev set for evaluation. The range of hyperparameter sets and optimal hyperparameters on the MHS, HF, and UB datasets are presented in Table 6.3, 6.4, and 6.5 respectively. We run all experiments with 5 random seeds and report the average results on the test sets. The models are trained on a single RTX A6000 GPU. We evaluate all models on the main task, toxicity, and report results in terms of precision, recall, and F1 for the minority class (TOX).

MHS				
Hyper-param	Range of values	STL	MTL	MTL-adv
LR	[1e-5, 2e-5, 3e-5, 5e-5, 1e-4]	1e-5	1e-5	5e-5
Dropout	[0.1, 0.15, 0.2, 0.25, 0.3]	0.15	0.1	0.2
Warmup	[0, 0.1, 0.2]	0.1	0.1	0
Scheduler	['constant', 'linear', 'cosine']	'constant'	'constant'	'constant'
Finetune_BERT_layer	[4, 8, 12]	12	12	12
α_1	[0.1, 0.25, 0.5, 0.75, 1.0]	N/A	N/A	0.75
α_2	[0.1, 0.25, 0.5, 0.75, 1.0]	N/A	N/A	0.75
Num_feature_extractor_layer	[1, 2, 3]	N/A	N/A	2
Freeze_BERT_for_ L_{disc} L_{adv}	[true, false]	N/A	N/A	true

Table 6.3: Hyperparameters tuned and the optimal values on the MHS dataset.

HF				
Hyper-param	Range of values	STL	MTL	MTL-adv
LR	[1e-5, 2e-5, 3e-5, 5e-5]	1e-5	1e-5	1e-5
Dropout	[0.1, 0.2, 0.3]	0.3	0.3	0.3
Warmup	[0, 0.1, 0.2]	0.1	0.1	0
Scheduler	['constant', 'linear']	'constant'	'linear'	'constant'
Finetune_BERT_layer	[8, 12]	12	8	12
α_1	[0.1, 0.25, 0.5, 0.75, 1.0]	N/A	N/A	1
α_2	[0.1, 0.25, 0.5, 0.75, 1.0]	N/A	N/A	0.25
Num_feature_extractor_layer	[1, 2, 3]	N/A	N/A	1
Freeze_BERT_for_ L_{disc} L_{adv}	[true]	N/A	N/A	true

Table 6.4: Hyperparameters tuned and the optimal values on the HF dataset.

UB				
Hyper-param	Range of values	STL	MTL	MTL-adv
LR	[1e-5, 3e-5, 5e-5]	1e-5	1e-5	1e-5
Dropout	[0.1, 0.2, 0.3]	0.2	0.2	0.2
Warmup	[0]	0	0	0
Scheduler	['constant']	'constant'	'constant'	'constant'
Finetune_BERT_layer	[12]	12	12	12
α_1	[0.1, 0.25, 0.5, 0.75]	N/A	N/A	0.5
α_2	[0.1, 0.25, 0.5, 0.75]	N/A	N/A	0.5
Num_feature_extractor_layer	[1, 2, 3]	N/A	N/A	3
Freeze_BERT_for_ L_{disc} L_{adv}	[true]	N/A	N/A	true

Table 6.5: Hyperparameters tuned and the optimal values on the UB dataset.

6.5 Debiasing Toxicity with Sentiment

We begin by comparing the models on the MHS dataset, with the sentiment task for all multi-task settings. Table 6.6 reports toxicity task results. First, the single-task model (STL) reaches a 0.745 F1 score on the test set, and vanilla multi-task training (MTL) improves precision but slightly worsens recall, resulting in a similar F1 score as the STL model. In contrast, the proposed adversarial MTL model (MTL-adv) improves both precision and recall over the STL and MTL baselines, leading to an improvement of 0.025 F1 score. Note that, as shown in Table 6.2, in the MHS dataset, positive and neutral examples are mostly non-toxic. We find that the MTL model enforces this bias and always predicts non-toxic when its sentiment head predicts a positive or neutral sentiment. Therefore, it may mis-classify toxic examples as non-toxic when the sentiment head incorrectly predicts positive or neutral sentiment, thus resulting in lower recall. However, the adversarial training in MTL-adv disentangles the toxicity and sentiment task representations and thus mitigates this bias.

To further validate the effect of our proposed method, we perform an ablation study by removing one loss at a time from the adversarial MTL setting. The bottom part of the Table 6.6 shows that removing the adversarial loss causes a large drop in recall, confirming the benefit of adversarial training for debiasing. Removing the discrimination loss and task-specific loss both result in higher precision and lower recall. Regarding the task-common loss, removing it decreases both precision and recall. This indicates that the task-common feature can benefit toxicity detection by learning from the sentiment task, rather than trading off precision and

Models	Precision	Recall	F1
STL	0.725	0.769	0.745
MTL	0.738	0.746	0.742
MTL-adv	0.734	0.811	0.770
w/o $\mathcal{L}_d$	0.748	0.771	0.759
w/o $\mathcal{L}_a$	0.756	0.754	0.754
w/o $\mathcal{L}_{cm}$	0.731	0.789	0.759
w/o $\mathcal{L}_s$	0.750	0.770	0.759

Table 6.6: Toxicity results on the MHS test set for the sentiment bias: the bottom part shows ablation experiments where we remove one loss at a time.

recall. Each of the ablated models exhibits lower performance when compared to MTL-adv while still outperforming the STL and MTL baselines.

Finally, in Figure 6.2 we visualize the common, toxicity-specific, and sentiment-specific features, by mapping feature vectors extracted from test set instances to a 2D space using T-SNE (van der Maaten and Hinton, 2008). We observe that (i) each feature type has its distinct cluster, and (ii) the clusters are well separated from each other. This shows that our model can successfully disentangle task-related information. We also notice that the three clusters have a similar shape, which indicates that the shared and private features still preserve semantic similarity across sentences (as they were extracted from the same examples).

6.6 Debiasing Toxicity with Profanity

Table 6.7 presents the results of debiasing with a profanity detection task. The three models perform well on the HF dataset, with very marginal differences. The MTL model shows a similar performance to the STL baseline, while the MTL-adv model underperforms the STL model by a small margin. These results are not surprising given that the annotation guidelines for the HF dataset instruct annotators to label content containing profanity words as toxic, and thus in most of the samples, toxicity and profanity correlate (Table 6.2).

Therefore, we also evaluate the model’s performance on the OOD test set containing a higher proportion of toxic examples without profanity words and non-toxic examples with profanity

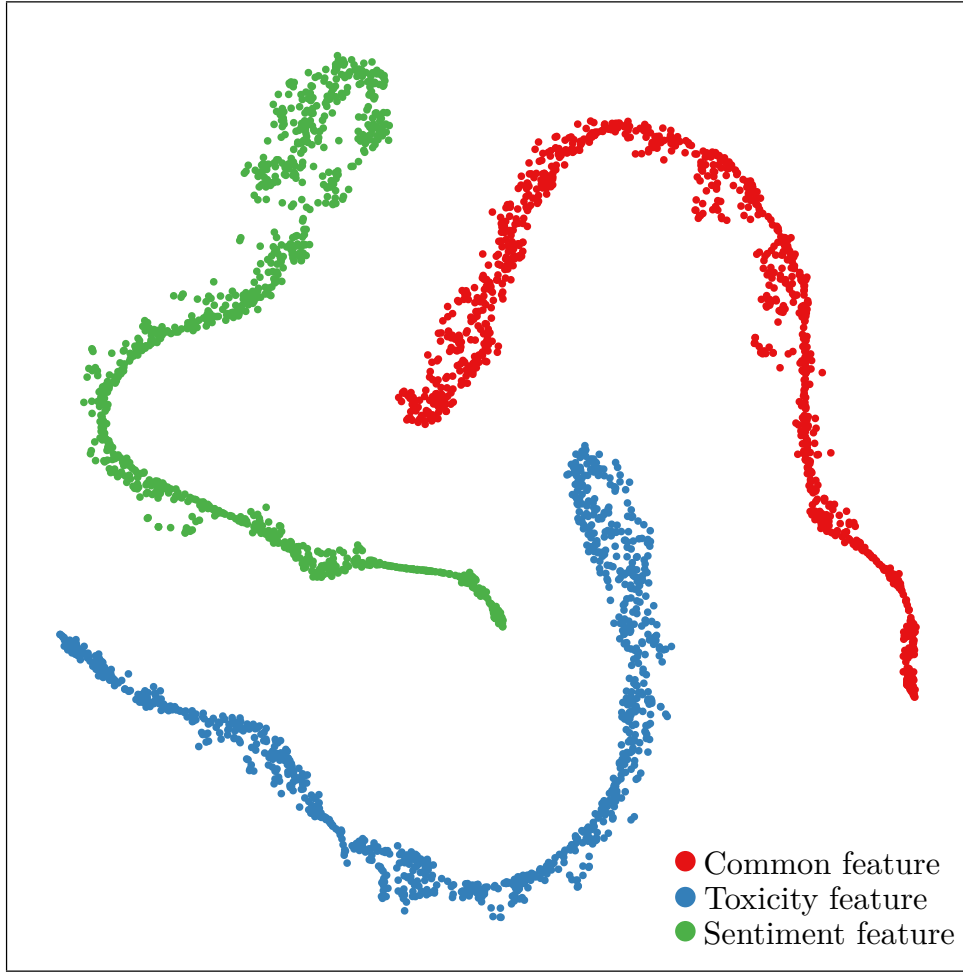


Figure 6.2: Visualization of the task-common/specific features using t-SNE (van der Maaten and Hinton, 2008). Each point represents a feature representation of a test set example from the MHS dataset.

words. The results are shown in Table 6.8. All models have much lower performance compared to the results on the in-domain HF test set. Comparatively, the MTL model outperforms the STL model on all metrics, and the MTL-adv model further improves the F1 score. This improvement is mainly from a substantial increase in recall, which means our proposed method can better capture implicit toxic samples than the STL and MTL baselines.

Finally, to confirm that the task-specific adversarial training works as expected, we evaluate the performance of the toxicity-specific feature (f_s^T) and the profanity-specific feature (f_s^P). Using the profanity feature results in the lowest performance, which confirms that the adversarial loss has achieved the desired result that the profanity-specific feature representation should focus on profanity and not toxicity. Using f_s^T only leads to better precision and F1, which further indicates that in this OOD setting, the profanity information is not helpful, and thus removing

Models	Precision	Recall	F1
STL	0.927	0.910	0.919
MTL	0.924	0.915	0.920
MTL-adv	0.909	0.918	0.914

Table 6.7: Toxicity results on the in-domain HF dataset.

it can be beneficial.

Models	Precision	Recall	F1
STL	0.524	0.447	0.482
MTL	0.533	0.483	0.504
MTL-adv	0.512	0.533	0.522
MTL-adv- f_s^T	0.585	0.506	0.542
MTL-adv- f_s^P	0.449	0.303	0.362

Table 6.8: Toxicity results on the OOD test set when trained on profanity-biased data.

To further probe the models, we follow Borkan et al. (2019) and perform a bias evaluation under three subsets of the OOD test set:

- **Subgroup**: examples where the bias task label is positive (i.e., with profanity), regardless of the main task label.
- **Background positive subgroup negative (BPSN)**: toxic examples with negative bias task label or non-toxic examples with positive bias task label (i.e., toxic examples without profanity or non-toxic examples with profanity). This setting measures whether the model learns the unintended bias between toxicity and the bias task.
- **Background negative subgroup positive (BNSP)**: toxic examples with positive bias task label or non-toxic examples with negative bias task label (i.e., toxic examples with profanity or non-toxic examples without profanity).

The bias evaluation results are in Table 6.9. We report the F1-TOX score under the three settings. Though trained on highly biased training data, MTL-adv shows a better performance than the other models on the OOD test set under both Subgroup and BPSN settings. In particular, MTL-adv improves the BPSN F1 substantially over the STL and MTL baselines,

which indicates the MTL-adv model is much less biased with the presence of profanity words. Under the BNSP setting, where the profanity label is equal to the toxicity label, unsurprisingly, the STL model shows the best performance because it is the most biased model and would correlate toxicity with profanity.

Models	F1		
	Subgroup	BPSN	BNSP
STL	0.609	0.405	0.603
MTL	0.615	0.452	0.590
MTL-adv	0.620	0.489	0.572

Table 6.9: Bias evaluation results on the OOD test set for the profanity bias.

To gain additional insight into our model’s behavior, we inspect how much the profanity words contribute to toxicity prediction through feature attribution. Specifically, we use Integrated Gradients (Sundararajan et al., 2017), a well known feature attribution method that estimates the importance of input features (words in our case) with respect to model predictions by computing the integral of gradients with respect to inputs along a path from a given baseline to the inputs. We select the 5 most frequent profanity words from the HF training set and report their average attribution values in Table 6.10. We find that the attribution values by the MTL-adv model are consistently lower than those for the STL or MTL models, which shows that our proposed MTL-adv model is less biased by profanity words when classifying toxicity.

	STL	MTL	MTL-adv
fucking	0.613	0.720	0.483
stupid	0.796	0.752	0.719
fuck	0.934	0.804	0.778
idiot	0.921	0.782	0.737
ass	0.805	0.752	0.539

Table 6.10: Average attribution values for top profanity words by the three models.

Furthermore, to probe what each model relies on when flagging toxicity, we follow the word importance ranking method commonly used in adversarial attacks as described in Section 2.2.2.1. Specifically, given a sentence $S = [w_1, w_2, \dots, w_n]$ of n words, we first compute its probability of being toxic by taking the model’s logit output $o(S)$. We then iteratively create an perturbed

sentence $S_{\setminus w_i} = [w_1, w_2, \dots, w_{i-1}, [\text{UNK}], w_{i+1}, \dots, w_n]$ by masking each word in the sentence with the [UNK] token. The toxicity probability $o(S_{\setminus w_i})$ of the perturbed sentence is then computed, and the difference between the original probability and the perturbed probability is considered as the importance of the word $I_{w_i} = \text{abs}(o(S) - o(S_{\setminus w_i}))$. Table 6.11 lists the 20 highest-scoring words per model. The single-task baseline (STL) is dominated by profanity: nearly every top-10 word is a slur or swear. Compared to the baselines, MTL-adv lowers both the frequency and the average score of profanity terms. For example, the importance score for “cucked” drops from 0.633 to 0.458 in MTL-adv. Taken together with the feature attribution analysis, these results confirm that MTL-adv succeeds in disentangling toxicity cues from profanity, thereby mitigating the underlying bias.

STL	I_{w_i}	MTL	I_{w_i}	MTL-adv	I_{w_i}
retards	0.808	retards	0.762	retards	0.821
raping	0.698	whores	0.616	idiots	0.495
cucked	0.633	cucked	0.521	brotherhood	0.481
idiots	0.630	moron	0.498	moron	0.468
whores	0.538	abolish	0.464	cucked	0.458
stupid	0.512	idiots	0.446	whores	0.433
moron	0.497	aryan	0.357	oppression	0.406
faggots	0.485	brotherhood	0.354	fat	0.381
shut	0.410	oppression	0.353	stupid	0.379
destroy	0.382	harvard	0.340	aryan	0.372
gassed	0.343	stupid	0.330	pathetic	0.341
pathetic	0.321	trumps	0.325	shut	0.288
domestic	0.316	kick	0.320	ethnostate	0.285
oppression	0.313	pathetic	0.309	lame	0.281
antiwhites	0.309	ruined	0.307	racist	0.280
nazis	0.305	sucks	0.294	negroes	0.274
sucks	0.303	boomer	0.293	rubbish	0.272
attacked	0.291	islam	0.291	outrageous	0.271
elitist	0.291	cuck	0.291	kkk	0.263
average	0.291	cucks	0.276	kick	0.260

Table 6.11: Most important words that contribute to a toxicity prediction by different models.

6.7 Debiasing Toxicity with Identities

The results for our third bias task – identity mention – on the UB dataset are shown in Table 6.12. The performance of STL, MTL, and MTL-adv models is similar in F1, with MTL-adv leading to lower precision but higher recall. The overall toxicity F1 score is not improving from the multi-task adversarial training, and this is caused by the data scarcity of the identity labels in the UB dataset. There are in total 24 identity labels in this dataset, but some identity type only has very few examples in the training set. For example, the UB training set contains over 1.4M examples, but only 905 of them contain positive labels for “latino”, and only 327 examples contain positive labels for “buddhist”.

Models	Precision	Recall	F1
STL	0.703	0.736	0.718
MTL	0.702	0.732	0.717
MTL-adv	0.685	0.746	0.714

Table 6.12: Toxicity results on the UB dataset for the identity bias.

Therefore, to understand whether the MTL-adv is debiased on identity mentions, it is essential to evaluate on the subset where identity labels are present. We again perform the bias evaluation under Subgroup, BPSN, and BNSP settings for the identity types. We report the bias metrics on 5 frequent identity groups for each identity type (gender, race, religion, sex orientation) and the average result on all 24 identities in Figure 6.3. Under the Subgroup setting, the MTL-adv model outperforms STL and MTL models in both average and individual identity scores. For the BPSN case, the improvements from using MTL-adv are very clear in this case, which indicates that our MTL-adv model is better at debiasing on identity terms. Finally, under the BNSP setting, all models show a worse performance. Such a drop in F1 score is caused by the data imbalance in the UB dataset, where the toxic examples with identity labels only account for 1.5% of the data, thus meaning that most predictions under the BNSP setting are non-toxic. Overall, the MTL-adv model shows a better performance under the Subgroup and BPSN settings, suggesting that the model could mitigate the unintended bias between toxicity and identity.

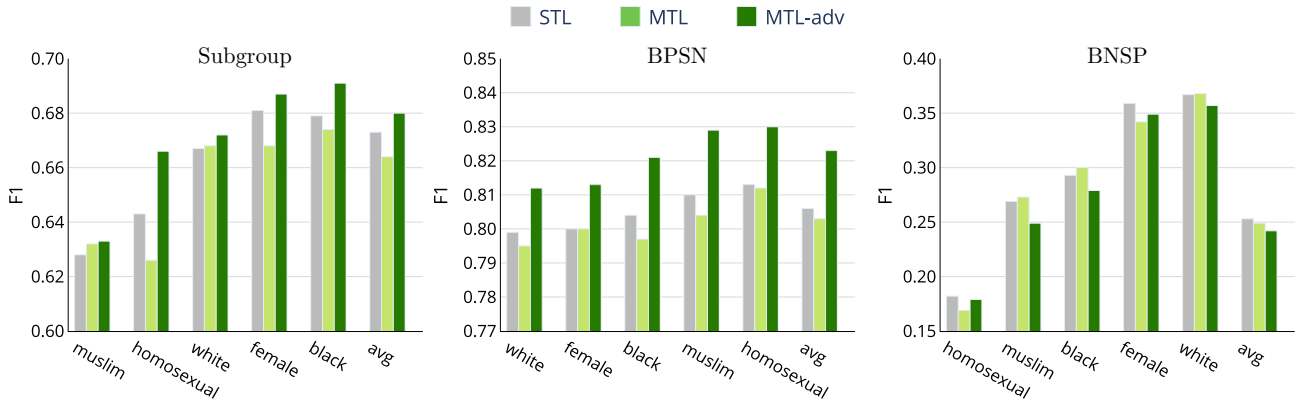


Figure 6.3: Bias evaluation results on UB dataset.

	STL	MTL	MTL-adv
muslim	0.162	0.184	0.177
homosexual	0.157	0.169	0.167
transgender	0.156	0.026	0.009
straight	0.149	0.056	0.076
blacks	0.147	0.154	0.132
white	0.139	0.143	0.119
gay	0.139	0.168	0.125
black	0.139	0.187	0.125
christian	0.125	0.107	0.073
islam	0.112	0.132	0.081
girls	0.094	0.084	0.092
Avg	0.132	0.123	0.103

Table 6.13: Top-10 identity words and their importance as estimated by the different models.

We then compute the average importance for identity words in the test set using the list of identity terms from Attanasio et al. (2022). We present the Top-10 identity words from the STL model and their importance scores in Table 6.13.

The MTL-adv model shows the lowest average importance on all frequent identities, and is clearly less biased on all words except “muslim” and “homosexual”. It is also worth noticing that for “transgender” the importance score from the MTL-adv model is nearly zero, meaning that the model is not biased on the word when predicting toxicity. The results further confirm that our adversarial training can help mitigate the identity mention as a potential bias.

6.8 Conclusions

This chapter studies the problem of unintended biases in toxicity detection. We introduce a new multi-task training method with adversarial learning to disentangle task-specific representations and multi-task training to learn a task-common representation that shares relevant information across tasks. Our method is flexible as it does not require a pre-defined set of bias terms and can generalize to multiple tasks. To show that, we conduct experiments on three tasks – sentiment, profanity, and identity mentions – and demonstrate that our method can help mitigate biases from different tasks and improve toxicity detection in out-of-domain settings while improving or maintaining the in-domain performance. This chapter answers the third research question on mitigating unintended bias. Our proposed multi-task adversarial learning framework can separate hidden representations for different tasks, where the spurious correlations are sourced from, to mitigate biases in toxicity detection models.

Although we took two tasks at a time in our experiments due to the lack of datasets with multi-task annotations, the proposed method can be applied to any number of related tasks. In future work, we will experiment with multi-task models over more diverse tasks and devise architectures that could improve and debias auxiliary tasks and the main task. Furthermore, alternative approaches for reducing biases, such as regularization-based methods, are complementary to ours and could further improve results.

Chapter 7

Conclusions

7.1 Summary of Achievements

In this thesis, we study the robustness of NLP models, specifically dealing with three robustness issues: textual noise, adversarial attacks, and unintended bias. This section provides an overall summary of this thesis, together with the key findings and achievements for each individual chapter, and how the research questions have been addressed.

RQ1: Can augmenting models with additional context signals improve model robustness to noise and perturbations?

In Chapter 3, we propose to incorporate visual cues to improve machine translation robustness against noisy texts. We combine adversarial training using error correction with multimodal MT and evaluate the robustness to common types of noise created by humans. Our experiments reveal that visual contexts can improve model robustness to both known and unseen noise. We further analyze the improved robustness from visual context and error correction training, showing that their effects are complementary and more powerful with noisier texts.

Further extending the exploration of textual noise, in Chapter 4 we look into leveraging extra visual and textual contexts to help handle noise. To do that, we propose a new challenging task, multimodal conversation derailment detection, and curate a new MCD dataset, which contains

texts, images, and conversations. In addition, we also propose a novel hierarchical architecture MMConv, which uses both visual and conversational contexts to model the conversation. Our experiments show that MMConv outperforms existing unimodal and multimodal architectures in this task. We then evaluate the model’s robustness, showing that both images and conversation can provide useful context to improve robustness, and they both contribute to the improved robustness complementarily. Both Chapter 3 and 4 help answer research question 1 and validate the positive effect of incorporating additional contextual signals, visual and/or contextual information, can help improve model robustness to noise and perturbations.

RQ2: Can a denoising or perturbation removal strategy generalize and defend against iterative adversarial perturbations at scale?

In Chapter 5, we address the issue of adversarial attacks on NLP models. This is more challenging, as the attacker can iteratively perturb inputs until an incorrect prediction from the target model. We propose a novel diffusion training method that incorporates a diffusion layer as a denoiser between the encoder and the classifier. At inference time, the diffusion layer iteratively removes noise from the encoder’s hidden states, producing a robust text representation even with adversarial input. The diffusion training can be applied to any model and can generalize to unseen adversarial attacks. Our systematic experiments show that the proposed method can achieve state-of-the-art robustness against common textual adversarial attacks, including black-box and white-box attacks. Chapter 5 answers the 2nd research question on tackling iterative attacks and provides a generalizable solution – iterative denoising – for different types of adversarial attacks.

RQ3: Beyond inference-time defenses, can representation-level disentanglement mitigate unintended biases by separating spurious task correlations during training?

Finally, in Chapter 6, we address the issue of unintended bias in toxicity detection. We propose a novel adversarial multitask method where potential biases can be modeled as additional tasks. The multi-task learning approach extracts representations that share relevant information across tasks and disentangles task-specific representations that could lead to bias with adversarial learning. We experiment with three potential biases in toxicity detection: senti-

ment, profanity, and identity mentions, and show that our method can successfully mitigate unintended biases and improve toxicity detection performance in out-of-domain settings while improving or maintaining performance on in-domain settings. Chapter 6 answers the last research question and shows that multi-task adversarial debias training can disentangle spurious correlations sourced from relevant tasks, and as a result mitigate unintended bias for the toxicity detection model.

Across this thesis, we study and understand how NLP systems can remain unreliable due to induced and learned vulnerabilities. Specifically, we focus on textual noise and adversarial attacks for the most common and severe induced noise. Both methods of using visual/conversational contexts and the diffusion-based denoising concentrate on the generalizable robustness to a variety of non-standard inputs. On the other hand, we explore an approach to improve model robustness on learned vulnerabilities, specifically on unintended bias during the training stage. The proposed technique sheds light on the path to less biased and unbiased toxicity detection models. Taken together, these findings form a coherent path towards robust and trustworthy NLP:

- Fuse complementary information (e.g., images, conversation history) to offset information loss from non-standard input;
- Combining adversarial training with latent noise removal can significantly improve robustness;
- Disentangle representation-level correlation with multi-task modeling.

7.2 Future Directions

7.2.1 Efficient and Robust NLP

Although current research has proposed various methods to improve the robustness of NLP models, they may introduce computational overhead that hinders real-time applications, either

at training or inference time. For example, adversarial training requires additional perturbed data during the training; denoising would increase the inference time. Therefore, it is important to explore efficient robustness techniques to improve NLP models without sacrificing costs. Current research focuses on efficient adversarial training via pruning (Xi et al., 2022), while efficient robustness methods from CV can be borrowed, e.g., partial adversarial training (Gupta et al., 2020), robustness distillation (Goldblum et al., 2020), etc.

7.2.2 LLM Red Teaming

This thesis mainly focused on improving the robustness of small pre-trained language models, however, it is also essential to explore the robustness of large language models. Even though LLMs are optimized with alignment training so that they are aligned with human values, they still face critical challenges in robustness, such as jailbreaks and prompt injections (Xu et al., 2024; Liu et al., 2024). Similar to adversarial attacks we discussed in Chapter 5, users can exploit LLMs by using different prompts to bypass their safeguards, leading them to produce unintended or harmful outputs or execute actions contrary to their original intent. Even though current SOTA LLMs such as ChatGPT are post-trained (Rafailov et al., 2024; Zhao et al., 2024) and prompted with guardrails (Dong et al., 2024), they can still be attacked by carefully curated inputs. For example, the DAN mode prompts for ChatGPT, which stands for "Do Anything Now", have been evolving by the effort from the community to jailbreak the model¹. Therefore, it is an important future research direction to study the area of red teaming in LLMs and how they can be built with enhanced robustness. To achieve a robust LLM system, safeguards from various perspectives might need to be combined together, e.g., robust pre-training via alignment (Bhardwaj and Poria, 2023), attack detection (Kim et al., 2024), input prompt rewriting (Xie et al., 2023), etc, which constitute the potential future research directions of LLM red teaming.

¹<https://gist.github.com/coolaj86/6f4f7b30129b0251f61fa7baaa881516>

7.2.3 Persistable Robustness in Models

In the era of large language models, pre-trained LLMs can be fine-tuned or prompted by downstream users to build NLP applications. Therefore, one significant challenge in NLP is ensuring that robustness persists through continual fine-tuning or transfer learning (Turbal et al., 2024). Qi et al. (2023) show that the safeguards in open-weight LLMs can be easily removed via parameter editing or fine-tuning. Such a kind of fine-tuning is considered as "harmful fine-tuning" or "tampering attack" (Huang et al., 2024). However, only few research has focused on this area in persistable safeguards for LLMs (Tamirisa et al., 2024), which leaves room for future research in building robust LLMs.

Bibliography

- Adams, C., Borkan, D., Inversion, Sorensen, J., Dixon, L., Vasserman, L., and Thain, N. (2019). Jigsaw unintended bias in toxicity classification. <https://kaggle.com/competitions/jigsaw-unintended-bias-in-toxicity-classification>. Kaggle.
- Al Sharou, K., Li, Z., and Specia, L. (2021). Towards a better understanding of noise in natural language processing. In *Proceedings of the International Conference on Recent Advances in Natural Language Processing (RANLP 2021)*, pages 53–62, Held Online. INCOMA Ltd.
- Alam, M. M. I. and Anastasopoulos, A. (2020). Fine-tuning MT systems for robustness to second-language speaker variations. In *Proceedings of the Sixth Workshop on Noisy User-generated Text (W-NUT 2020)*, pages 149–158, Online. Association for Computational Linguistics.
- Almerekhi, H., Jansen, S. b. B. J., and Kwak, c.-s. b. H. (2020). *Investigating Toxicity Across Multiple Reddit Communities, Users, and Moderators*, page 294–298. Association for Computing Machinery, New York, NY, USA.
- Alzantot, M., Sharma, Y., Elgohary, A., Ho, B.-J., Srivastava, M., and Chang, K.-W. (2018). Generating natural language adversarial examples. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2890–2896, Brussels, Belgium. Association for Computational Linguistics.
- Anastasopoulos, A., Lui, A., Nguyen, T. Q., and Chiang, D. (2019). Neural machine translation of text from non-native speakers. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Tech-*

nologies, Volume 1 (Long and Short Papers), pages 3070–3080, Minneapolis, Minnesota. Association for Computational Linguistics.

Anderson, P., He, X., Buehler, C., Teney, D., Johnson, M., Gould, S., and Zhang, L. (2018a). Bottom-up and top-down attention for image captioning and visual question answering. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6077–6086.

Anderson, P., He, X., Buehler, C., Teney, D., Johnson, M., Gould, S., and Zhang, L. (2018b). Bottom-up and top-down attention for image captioning and visual question answering. In *CVPR*.

Anthropic (2024). The claude 3 model family: Opus, sonnet, haiku.

Attanasio, G., Nozza, D., Hovy, D., and Baralis, E. (2022). Entropy-based attention regularization frees unintended bias mitigation from lists. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 1105–1119, Dublin, Ireland. Association for Computational Linguistics.

Ba, J. L., Kiros, J. R., and Hinton, G. E. (2016). Layer normalization.

Bahdanau, D., Cho, K., and Bengio, Y. (2015). Neural machine translation by jointly learning to align and translate. In *ICLR 2015*.

Balakrishnan, S., Fang, Y., and Zhu, X. (2022). Exploring robustness of prefix tuning in noisy data: A case study in financial sentiment analysis. In Chen, C.-C., Huang, H.-H., Takamura, H., and Chen, H.-H., editors, *Proceedings of the Fourth Workshop on Financial Technology and Natural Language Processing (FinNLP)*, pages 78–88, Abu Dhabi, United Arab Emirates (Hybrid). Association for Computational Linguistics.

Barrault, L., Bougares, F., Specia, L., Lala, C., Elliott, D., and Frank, S. (2018). Findings of the third shared task on multimodal machine translation. In *Proceedings of the Third Conference on Machine Translation: Shared Task Papers*, pages 304–323, Belgium, Brussels. Association for Computational Linguistics.

- Bauer, L., Gopalakrishnan, K., Gella, S., Liu, Y., Bansal, M., and Hakkani-Tur, D. (2021). Analyzing the limits of self-supervision in handling bias in language.
- Baumgartner, J., Zannettou, S., Keegan, B., Squire, M., and Blackburn, J. (2020). The pushshift reddit dataset.
- Baziotis, C., Pelekis, N., and Doukeridis, C. (2017). Datastories at semeval-2017 task 4: Deep lstm with attention for message-level and topic-based sentiment analysis. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, pages 747–754, Vancouver, Canada. Association for Computational Linguistics.
- Belinkov, Y. and Bisk, Y. (2018). Synthetic and natural noise both break neural machine translation. In *International Conference on Learning Representations*.
- Beltagy, I., Peters, M. E., and Cohan, A. (2020). Longformer: The long-document transformer.
- Berard, A., Calapodescu, I., and Roux, C. (2019). Naver labs Europe’s systems for the WMT19 machine translation robustness task. In *Proceedings of the Fourth Conference on Machine Translation (Volume 2: Shared Task Papers, Day 1)*, pages 526–532, Florence, Italy. Association for Computational Linguistics.
- Bhardwaj, R. and Poria, S. (2023). Red-teaming large language models using chain of utterances for safety-alignment.
- Blodgett, S. L., Barocas, S., Daumé III, H., and Wallach, H. (2020). Language (technology) is power: A critical survey of “bias” in NLP. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5454–5476, Online. Association for Computational Linguistics.
- Bolding, Q., Liao, B., Denis, B., Luo, J., and Monz, C. (2023). Ask language model to clean your noisy translation data. In Bouamor, H., Pino, J., and Bali, K., editors, *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 3215–3236, Singapore. Association for Computational Linguistics.

- Bolukbasi, T., Chang, K.-W., Zou, J., Saligrama, V., and Kalai, A. (2016). Man is to computer programmer as woman is to homemaker? debiasing word embeddings. In *Proceedings of the 30th International Conference on Neural Information Processing Systems*, NIPS'16, page 4356–4364, Red Hook, NY, USA. Curran Associates Inc.
- Bommasani, R., Hudson, D. A., Adeli, E., Altman, R., Arora, S., von Arx, S., Bernstein, M. S., Bohg, J., Bosselut, A., Brunskill, E., Brynjolfsson, E., Buch, S., Card, D., Castellon, R., Chatterji, N., Chen, A., Creel, K., Davis, J. Q., Demszky, D., Donahue, C., Doumbouya, M., Durmus, E., Ermon, S., Etchemendy, J., Ethayarajh, K., Fei-Fei, L., Finn, C., Gale, T., Gillespie, L., Goel, K., Goodman, N., Grossman, S., Guha, N., Hashimoto, T., Henderson, P., Hewitt, J., Ho, D. E., Hong, J., Hsu, K., Huang, J., Icard, T., Jain, S., Jurafsky, D., Kalluri, P., Karamcheti, S., Keeling, G., Khani, F., Khattab, O., Koh, P. W., Krass, M., Krishna, R., Kuditipudi, R., Kumar, A., Ladhak, F., Lee, M., Lee, T., Leskovec, J., Levent, I., Li, X. L., Li, X., Ma, T., Malik, A., Manning, C. D., Mirchandani, S., Mitchell, E., Munyikwa, Z., Nair, S., Narayan, A., Narayanan, D., Newman, B., Nie, A., Niebles, J. C., Nilforoshan, H., Nyarko, J., Ogut, G., Orr, L., Papadimitriou, I., Park, J. S., Piech, C., Portelance, E., Potts, C., Raghunathan, A., Reich, R., Ren, H., Rong, F., Roohani, Y., Ruiz, C., Ryan, J., Ré, C., Sadigh, D., Sagawa, S., Santhanam, K., Shih, A., Srinivasan, K., Tamkin, A., Taori, R., Thomas, A. W., Tramèr, F., Wang, R. E., Wang, W., Wu, B., Wu, J., Wu, Y., Xie, S. M., Yasunaga, M., You, J., Zaharia, M., Zhang, M., Zhang, T., Zhang, X., Zhang, Y., Zheng, L., Zhou, K., and Liang, P. (2021). On the opportunities and risks of foundation models.
- Borkan, D., Dixon, L., Sorensen, J., Thain, N., and Vasserman, L. (2019). Nuanced metrics for measuring unintended bias with real data for text classification. In *Companion Proceedings of The 2019 World Wide Web Conference*, WWW '19, page 491–500, New York, NY, USA. Association for Computing Machinery.
- Bose, T., Aletras, N., Illina, I., and Fohr, D. (2022). Dynamically refined regularization for improving cross-corpora hate speech detection. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 372–382, Dublin, Ireland. Association for Computational Linguistics.

- Bouritsas, G., Koutras, P., Zlatintsi, A., and Maragos, P. (2018). Multimodal visual concept learning with weakly supervised techniques. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.
- Bousmalis, K., Trigeorgis, G., Silberman, N., Krishnan, D., and Erhan, D. (2016). Domain separation networks. In *Proceedings of the 30th International Conference on Neural Information Processing Systems, NIPS'16*, page 343–351, Red Hook, NY, USA. Curran Associates Inc.
- Bryant, C., Felice, M., and Briscoe, T. (2017). Automatic annotation and evaluation of error types for grammatical error correction. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 793–805, Vancouver, Canada. Association for Computational Linguistics.
- Burchardt, A. (2013). Multidimensional quality metrics: a flexible system for assessing translation quality. In *Proceedings of Translating and the Computer 35*, London, UK. Aslib.
- Caglayan, O., Barrault, L., and Bougares, F. (2016). Multimodal attention for neural machine translation.
- Caglayan, O., Ive, J., Haralampieva, V., Madhyastha, P., Barrault, L., and Specia, L. (2020). Simultaneous machine translation with visual context. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2350–2361, Online. Association for Computational Linguistics.
- Caglayan, O., Madhyastha, P., Specia, L., and Barrault, L. (2019). Probing the need for visual context in multimodal machine translation. 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 4159–4170, Minneapolis, Minnesota. Association for Computational Linguistics.
- Cai, Y. and Wan, X. (2019). Multi-domain sentiment classification based on domain-aware embedding and attention. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI-19*, pages 4904–4910. International Joint Conferences on Artificial Intelligence Organization.

- Calixto, I., Elliott, D., and Frank, S. (2016). DCU-UvA multimodal MT system report. In *Proceedings of the First Conference on Machine Translation: Volume 2, Shared Task Papers*, pages 634–638, Berlin, Germany. Association for Computational Linguistics.
- Calixto, I. and Liu, Q. (2017). Incorporating global visual features into attention-based neural machine translation. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 992–1003, Copenhagen, Denmark. Association for Computational Linguistics.
- Calixto, I., Liu, Q., and Campbell, N. (2017). Doubly-attentive decoder for multi-modal neural machine translation. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1913–1924, Vancouver, Canada. Association for Computational Linguistics.
- Calixto, I., Rios, M., and Aziz, W. (2019). Latent variable model for multi-modal translation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 6392–6405, Florence, Italy. Association for Computational Linguistics.
- Carlini, N. and Wagner, D. (2017). Towards evaluating the robustness of neural networks.
- Cécillon, N., Labatut, V., Dufour, R., and Linarès, G. (2020). WAC: A corpus of Wikipedia conversations for online abuse detection. In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 1382–1390, Marseille, France. European Language Resources Association.
- Cer, D., Yang, Y., Kong, S.-y., Hua, N., Limtiaco, N., St. John, R., Constant, N., Guajardo-Cespedes, M., Yuan, S., Tar, C., Strope, B., and Kurzweil, R. (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.
- Chang, J. P. and Danescu-Niculescu-Mizil, C. (2019). Trouble on the horizon: Forecasting the derailment of online conversations as they develop. 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)*, pages 4743–4754, Hong Kong, China. Association for Computational Linguistics.
- Chang, M.-W., Toutanova, K., Lee, K., and Devlin, J. (2019). Language model pre-training for hierarchical document representations.
- Chen, X. and Cardie, C. (2018). Multinomial adversarial networks for multi-domain text classification. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1226–1240, New Orleans, Louisiana. Association for Computational Linguistics.
- Chen, X., He, B., Hui, K., Sun, L., and Sun, Y. (2023). Dealing with textual noise for robust and effective bert re-ranking. *Inf. Process. Manage.*, 60(1).
- Cheng, J. and Amiri, H. (2024). FairFlow: Mitigating dataset biases through undecided learning for natural language understanding. In Al-Onaizan, Y., Bansal, M., and Chen, Y.-N., editors, *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 21960–21975, Miami, Florida, USA. Association for Computational Linguistics.
- Cheng, X., Luo, D., Chen, X., Liu, L., Zhao, D., and Yan, R. (2023). Lift yourself up: Retrieval-augmented text generation with self-memory. In *Thirty-seventh Conference on Neural Information Processing Systems*.
- Cheng, Y., Jiang, L., and Macherey, W. (2019). Robust neural machine translation with doubly adversarial inputs. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4324–4333, Florence, Italy. Association for Computational Linguistics.
- Cheng, Y., Jiang, L., Macherey, W., and Eisenstein, J. (2020). AdvAug: Robust adversarial augmentation for neural machine translation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5961–5970, Online. Association for Computational Linguistics.

- Cheng, Y., Tu, Z., Meng, F., Zhai, J., and Liu, Y. (2018). Towards robust neural machine translation. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1756–1766, Melbourne, Australia. Association for Computational Linguistics.
- Cho, K., van Merriënboer, B., Bahdanau, D., and Bengio, Y. (2014a). On the properties of neural machine translation: Encoder–decoder approaches. In *Proceedings of SSST-8, Eighth Workshop on Syntax, Semantics and Structure in Statistical Translation*, pages 103–111, Doha, Qatar. Association for Computational Linguistics.
- Cho, K., van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., and Bengio, Y. (2014b). Learning phrase representations using RNN encoder–decoder for statistical machine translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1724–1734, Doha, Qatar. Association for Computational Linguistics.
- Churchill, E. F. and Bly, S. (2000). Culture vultures: Considering culture and communication in virtual environments. *SIGGROUP Bull.*, 21(1):6–11.
- Clark, K., Luong, M.-T., Le, Q. V., and Manning, C. D. (2020). ELECTRA: Pre-training text encoders as discriminators rather than generators. In *ICLR*.
- Cooper Stickland, A., Sengupta, S., Krone, J., Mansour, S., and He, H. (2023). Robustification of multilingual language models to real-world noise in crosslingual zero-shot settings with robust contrastive pretraining. In Vlachos, A. and Augenstein, I., editors, *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, pages 1375–1391, Dubrovnik, Croatia. Association for Computational Linguistics.
- Dahlmeier, D. and Ng, H. T. (2012). Better evaluation for grammatical error correction. In *Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 568–572, Montréal, Canada. Association for Computational Linguistics.

- Dao, T., Fu, D. Y., Ermon, S., Rudra, A., and Ré, C. (2022). FlashAttention: Fast and memory-efficient exact attention with IO-awareness. In *Advances in Neural Information Processing Systems (NeurIPS)*.
- Davis, M. (2003). Aoccdrnig to a rscheearch at cmabrigde uinervtisy.
- de Vassimon Manela, D., Errington, D., Fisher, T., van Breugel, B., and Minervini, P. (2021). Stereotype and skew: Quantifying gender bias in pre-trained and fine-tuned language models. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 2232–2242, Online. Association for Computational Linguistics.
- Denkowski, M. and Lavie, A. (2014). Meteor universal: Language specific translation evaluation for any target language. In *Proceedings of the Ninth Workshop on Statistical Machine Translation*, pages 376–380, Baltimore, Maryland, USA. Association for Computational Linguistics.
- Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. (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, Minneapolis, Minnesota. Association for Computational Linguistics.
- Dhariwal, P. and Nichol, A. Q. (2021). Diffusion models beat GANs on image synthesis. In Beygelzimer, A., Dauphin, Y., Liang, P., and Vaughan, J. W., editors, *Advances in Neural Information Processing Systems*.
- Dhole, K., Gangal, V., Gehrmann, S., Gupta, A., Li, Z., Mahamood, S., Mahadiran, A., Mille, S., Shrivastava, A., Tan, S., Wu, T., Sohl-Dickstein, J., Choi, J., Hovy, E., Dušek, O., Ruder, S., Anand, S., Aneja, N., Banjade, R., Barthe, L., Behnke, H., Berlot-Attwell, I., Boyle, C., Brun, C., Cabezudo, M. A. S., Cahyawijaya, S., Chapuis, E., Che, W., Choudhary, M., Clauss, C., Colombo, P., Cornell, F., Dagan, G., Das, M., Dixit, T., Dopierre, T., Dray, P.-A., Dubey, S., Ekeinhor, T., Giovanni, M. D., Goyal, T., Gupta, R., Hamla, L., Han, S.,

Harel-Canada, F., Honoré, A., Jindal, I., Joniak, P., Kleyko, D., Kovatchev, V., Krishna, K., Kumar, A., Langer, S., Lee, S. R., Levinson, C. J., Liang, H., Liang, K., Liu, Z., Lukyanenko, A., Marivate, V., de Melo, G., Meoni, S., Meyer, M., Mir, A., Moosavi, N. S., Meunnighoff, N., Mun, T. S. H., Murray, K., Namysl, M., Obedkova, M., Oli, P., Pasricha, N., Pfister, J., Plant, R., Prabhu, V., Pais, V., Qin, L., Raji, S., Rajpoot, P. K., Raunak, V., Rinberg, R., Roberts, N., Rodriguez, J. D., Roux, C., Samus, V., Sai, A., Schmidt, R., Scialom, T., Sefara, T., Shamsi, S., Shen, X., Shi, Y., Shi, H., Shvets, A., Siegel, N., Sileo, D., Simon, J., Singh, C., Sitelew, R., Soni, P., Sorensen, T., Soto, W., Srivastava, A., Srivatsa, A., Sun, T., Varma, M., Tabassum, A., Tan, F., Teehan, R., Tiwari, M., Tolkiehn, M., Wang, A., Wang, Z., Wang, Z., Wang, G., Wei, F., Wilie, B., Winata, G. I., Wu, X., Wydmanski, W., Xie, T., Yaseen, U., Yee, M., Zhang, J., and Zhang, Y. (2023). NL-augmenter: A framework for task-sensitive natural language augmentation. *Northern European Journal of Language Technology*, 9.

Dhole, K. D., Gangal, V., Gehrmann, S., Gupta, A., Li, Z., Mahamood, S., Mahendiran, A., Mille, S., Srivastava, A., Tan, S., Wu, T., Sohl-Dickstein, J., Choi, J. D., Hovy, E., Dusek, O., Ruder, S., Anand, S., Aneja, N., Banjade, R., Barthe, L., Behnke, H., Berlot-Attwell, I., Boyle, C., Brun, C., Cabezudo, M. A. S., Cahyawijaya, S., Chapuis, E., Che, W., Choudhary, M., Clauss, C., Colombo, P., Cornell, F., Dagan, G., Das, M., Dixit, T., Dopierre, T., Dray, P.-A., Dubey, S., Ekeinhor, T., Di Giovanni, M., Gupta, R., Gupta, R., Hamla, L., Han, S., Harel-Canada, F., Honore, A., Jindal, I., Joniak, P. K., Kleyko, D., Kovatchev, V., Krishna, K., Kumar, A., Langer, S., Lee, S. R., Levinson, C. J., Liang, H., Liang, K., Liu, Z., Lukyanenko, A., Marivate, V., de Melo, G., Meoni, S., Meyer, M., Mir, A., Moosavi, N. S., Muennighoff, N., Mun, T. S. H., Murray, K., Namysl, M., Obedkova, M., Oli, P., Pasricha, N., Pfister, J., Plant, R., Prabhu, V., Pais, V., Qin, L., Raji, S., Rajpoot, P. K., Raunak, V., Rinberg, R., Roberts, N., Rodriguez, J. D., Roux, C., S., V. P. H., Sai, A. B., Schmidt, R. M., Scialom, T., Sefara, T., Shamsi, S. N., Shen, X., Shi, H., Shi, Y., Shvets, A., Siegel, N., Sileo, D., Simon, J., Singh, C., Sitelew, R., Soni, P., Sorensen, T., Soto, W., Srivastava, A., Srivatsa, K. A., Sun, T., T, M. V., Tabassum, A., Tan, F. A., Teehan, R., Tiwari, M., Tolkiehn, M., Wang, A., Wang, Z., Wang, G., Wang, Z. J., Wei, F., Wilie, B., Winata, G. I.,

- Wu, X., Wydmański, W., Xie, T., Yaseen, U., Yee, M., Zhang, J., and Zhang, Y. (2021). Nl-augmenter: A framework for task-sensitive natural language augmentation.
- Dixon, L., Li, J., Sorensen, J., Thain, N., and Vasserman, L. (2018). Measuring and mitigating unintended bias in text classification. In *Proceedings of the 2018 AAAI/ACM Conference on AI, Ethics, and Society*, AIES '18, page 67–73, New York, NY, USA. Association for Computing Machinery.
- Dong, L., Xu, S., and Xu, B. (2018). Speech-transformer: A no-recurrence sequence-to-sequence model for speech recognition. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5884–5888.
- Dong, Y., Mu, R., Jin, G., Qi, Y., Hu, J., Zhao, X., Meng, J., Ruan, W., and Huang, X. (2024). Building guardrails for large language models.
- Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., Uszkoreit, J., and Houlsby, N. (2021). An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*.
- Du, C., Sun, H., Wang, J., Qi, Q., and Liao, J. (2020). Adversarial and domain-aware BERT for cross-domain sentiment analysis. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4019–4028, Online. Association for Computational Linguistics.
- Du, L., Ding, X., Sun, Z., Liu, T., Qin, B., and Liu, J. (2023). Towards stable natural language understanding via information entropy guided debiasing. In Rogers, A., Boyd-Graber, J., and Okazaki, N., editors, *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2868–2882, Toronto, Canada. Association for Computational Linguistics.
- Ebrahimi, J., Lowd, D., and Dou, D. (2018a). On adversarial examples for character-level neural machine translation. In *Proceedings of the 27th International Conference on Computational*

- Linguistics*, pages 653–663, Santa Fe, New Mexico, USA. Association for Computational Linguistics.
- Ebrahimi, J., Lowd, D., and Dou, D. (2018b). On adversarial examples for character-level neural machine translation. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 653–663, Santa Fe, New Mexico, USA. Association for Computational Linguistics.
- Ebrahimi, J., Rao, A., Lowd, D., and Dou, D. (2018c). HotFlip: White-box adversarial examples for text classification. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 31–36, Melbourne, Australia. Association for Computational Linguistics.
- Elazar, Y. and Goldberg, Y. (2018). Adversarial removal of demographic attributes from text data. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 11–21, Brussels, Belgium. Association for Computational Linguistics.
- Elliott, D. (2018). Adversarial evaluation of multimodal machine translation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2974–2978, Brussels, Belgium. Association for Computational Linguistics.
- Elliott, D., Frank, S., Barrault, L., Bougares, F., and Specia, L. (2017a). Findings of the second shared task on multimodal machine translation and multilingual image description. In *Proceedings of the Second Conference on Machine Translation, Volume 2: Shared Task Papers*, pages 215–233, Copenhagen, Denmark. Association for Computational Linguistics.
- Elliott, D., Frank, S., Barrault, L., Bougares, F., and Specia, L. (2017b). Findings of the second shared task on multimodal machine translation and multilingual image description. In *Proceedings of the Second Conference on Machine Translation*, pages 215–233, Copenhagen, Denmark. Association for Computational Linguistics.
- Elliott, D., Frank, S., Sima'an, K., and Specia, L. (2016). Multi30K: Multilingual English-German image descriptions. In *Proceedings of the 5th Workshop on Vision and Language*, pages 70–74, Berlin, Germany. Association for Computational Linguistics.

- ElSherief, M., Ziems, C., Muchlinski, D., Anupindi, V., Seybolt, J., De Choudhury, M., and Yang, D. (2021). Latent hatred: A benchmark for understanding implicit hate speech. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 345–363, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.
- Fortuna, P. and Nunes, S. (2018a). A survey on automatic detection of hate speech in text. *ACM Comput. Surv.*, 51(4).
- Fortuna, P. and Nunes, S. (2018b). A survey on automatic detection of hate speech in text. *ACM Comput. Surv.*, 51(4).
- Founta, A., Djouvas, C., Chatzakou, D., Leontiadis, I., Blackburn, J., Stringhini, G., Vakali, A., Sirivianos, M., and Kourtellis, N. (2018). Large scale crowdsourcing and characterization of twitter abusive behavior. *Proceedings of the International AAAI Conference on Web and Social Media*, 12(1).
- Gan, W. C. and Ng, H. T. (2019). Improving the robustness of question answering systems to question paraphrasing. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 6065–6075, Florence, Italy. Association for Computational Linguistics.
- Ganin, Y. and Lempitsky, V. (2015). Unsupervised domain adaptation by backpropagation. In *Proceedings of the 32nd International Conference on International Conference on Machine Learning - Volume 37, ICML’15*, page 1180–1189. JMLR.org.
- Gao, J., Lanchantin, J., Soffa, M. L., and Qi, Y. (2018). Black-box generation of adversarial text sequences to evade deep learning classifiers. In *2018 IEEE Security and Privacy Workshops (SPW)*, pages 50–56.
- Gao, S., Dou, S., Liu, Y., Wang, X., Zhang, Q., Wei, Z., Ma, J., and Shan, Y. (2023). DSRM: Boost textual adversarial training with distribution shift risk minimization. In Rogers, A., Boyd-Graber, J., and Okazaki, N., editors, *Proceedings of the 61st Annual Meeting of the*

- Association for Computational Linguistics (Volume 1: Long Papers)*, pages 12177–12189, Toronto, Canada. Association for Computational Linguistics.
- Garg, S. and Ramakrishnan, G. (2020). BAE: BERT-based adversarial examples for text classification. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6174–6181, Online. Association for Computational Linguistics.
- Garg, T., Masud, S., Suresh, T., and Chakraborty, T. (2023). Handling bias in toxic speech detection: A survey. *ACM Comput. Surv.*, 55(13s).
- Garnier, M. and Saint-Dizier, P. (2016). Error typology and remediation strategies for requirements written in English by non-native speakers. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC’16)*, pages 190–197, Portorož, Slovenia. European Language Resources Association (ELRA).
- Goldblum, M., Fowl, L., Feizi, S., and Goldstein, T. (2020). Adversarially robust distillation. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(04):3996–4003.
- Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y. (2014a). Generative adversarial nets. In *Advances in neural information processing systems*, pages 2672–2680.
- Goodfellow, I. J., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y. (2014b). Generative adversarial networks.
- Gorrell, G., Petrak, J., and Bontcheva, K. (2015). Using @twitter conventions to improve #lod-based named entity disambiguation. In Gandon, F., Sabou, M., Sack, H., d’Amato, C., Cudré-Mauroux, P., and Zimmermann, A., editors, *The Semantic Web. Latest Advances and New Domains*, pages 171–186, Cham. Springer International Publishing.
- Gulsen, M., Cengiz, B., Sahin, Y. H., and Unal, G. (2024). PclD: Point cloud layerwise diffusion for adversarial purification.
- Guo, C., Sablayrolles, A., Jégou, H., and Kiela, D. (2021). Gradient-based adversarial attacks against text transformers. In *Proceedings of the 2021 Conference on Empirical Methods in*

- Natural Language Processing*, pages 5747–5757, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.
- Gupta, A., Blum, C. W., Choji, T., Fei, Y., Shah, S., Vempala, A., and Srikumar, V. (2023). Don't retrain, just rewrite: Countering adversarial perturbations by rewriting text. *arXiv preprint arXiv:2305.16444*.
- Gupta, S., Dube, P., and Verma, A. (2020). Improving the affordability of robustness training for dnns.
- Gururangan, S., Marasović, A., Swayamdipta, S., Lo, K., Beltagy, I., Downey, D., and Smith, N. A. (2020). Don't stop pretraining: Adapt language models to domains and tasks. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8342–8360, Online. Association for Computational Linguistics.
- Halevy, M., Harris, C., Bruckman, A., Yang, D., and Howard, A. (2021). Mitigating racial biases in toxic language detection with an equity-based ensemble framework. In *Equity and Access in Algorithms, Mechanisms, and Optimization*, EAAMO '21, New York, NY, USA. Association for Computing Machinery.
- Hartvigsen, T., Gabriel, S., Palangi, H., Sap, M., Ray, D., and Kamar, E. (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.
- He, K., Zhang, X., Ren, S., and Sun, J. (2015). Deep residual learning for image recognition.
- He, K., Zhang, X., Ren, S., and Sun, J. (2016). Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778.
- He, P., Gao, J., and Chen, W. (2023). Debertav3: Improving deberta using electra-style pre-training with gradient-disentangled embedding sharing.

- Heigold, G., Varanasi, S., Neumann, G., and van Genabith, J. (2018). How robust are character-based word embeddings in tagging and MT against word scrambling or random noise? In *Proceedings of the 13th Conference of the Association for Machine Translation in the Americas (Volume 1: Research Track)*, pages 68–80, Boston, MA. Association for Machine Translation in the Americas.
- Hendrycks, D. and Gimpel, K. (2016). Gaussian error linear units (gelus).
- Ho, J., Jain, A., and Abbeel, P. (2020). Denoising diffusion probabilistic models. In *Proceedings of the 34th International Conference on Neural Information Processing Systems, NIPS '20*, Red Hook, NY, USA. Curran Associates Inc.
- Hochreiter, S. and Schmidhuber, J. (1997a). Long Short-Term Memory. *Neural Computation*, 9(8):1735–1780.
- Hochreiter, S. and Schmidhuber, J. (1997b). Long short-term memory. *Neural Comput.*, 9(8):1735–1780.
- Huang, K.-H. and Chang, K.-W. (2021). Generating syntactically controlled paraphrases without using annotated parallel pairs. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 1022–1033, Online. Association for Computational Linguistics.
- Huang, R., Zou, B., Hong, Y., Zhang, W., Aw, A., and Zhou, G. (2020). NUT-RC: Noisy user-generated text-oriented reading comprehension. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 2687–2698, Barcelona, Spain (Online). International Committee on Computational Linguistics.
- Huang, T., Hu, S., Ilhan, F., Tekin, S. F., and Liu, L. (2024). Harmful fine-tuning attacks and defenses for large language models: A survey.
- Im, D. J., Ahn, S., Memisevic, R., and Bengio, Y. (2016). Denoising criterion for variational auto-encoding framework.

- Imankulova, A., Kaneko, M., Hirasawa, T., and Komachi, M. (2020). Towards multimodal simultaneous neural machine translation. In *Proceedings of the Fifth Conference on Machine Translation*, pages 592–601, Online. Association for Computational Linguistics.
- Ive, J., Anuchitanukul, A., and Specia, L. (2021). Revisiting contextual toxicity detection in conversations.
- Ive, J., Madhyastha, P., and Specia, L. (2019). Distilling translations with visual awareness. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 6525–6538, Florence, Italy. Association for Computational Linguistics.
- Ji, J., Hou, B., Zhang, Z., Zhang, G., Fan, W., Li, Q., Zhang, Y., Liu, G., Liu, S., and Chang, S. (2024). Advancing the robustness of large language models through self-denoised smoothing. In Duh, K., Gomez, H., and Bethard, S., editors, *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 2: Short Papers)*, pages 246–257, Mexico City, Mexico. Association for Computational Linguistics.
- Jia, R., Raghunathan, A., Göksel, K., and Liang, P. (2019). Certified robustness to adversarial word substitutions. 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)*, pages 4129–4142, Hong Kong, China. Association for Computational Linguistics.
- Jiang, H., He, P., Chen, W., Liu, X., Gao, J., and Zhao, T. (2020). SMART: Robust and efficient fine-tuning for pre-trained natural language models through principled regularized optimization. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 2177–2190, Online. Association for Computational Linguistics.
- Jin, D., Jin, Z., Zhou, J. T., and Szolovits, P. (2020). Is bert really robust? a strong baseline for natural language attack on text classification and entailment. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(05):8018–8025.

- Kaneko, M. and Bollegala, D. (2021). Debiasing pre-trained contextualised embeddings. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 1256–1266, Online. Association for Computational Linguistics.
- Karimi Mahabadi, R., Belinkov, Y., and Henderson, J. (2020). End-to-end bias mitigation by modelling biases in corpora. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8706–8716, Online. Association for Computational Linguistics.
- Karpukhin, V., Levy, O., Eisenstein, J., and Ghazvininejad, M. (2019). Training on synthetic noise improves robustness to natural noise in machine translation. In *Proceedings of the 5th Workshop on Noisy User-generated Text (W-NUT 2019)*, pages 42–47, Hong Kong, China. Association for Computational Linguistics.
- Kementchedjhieva, Y. and Søgaard, A. (2021). Dynamic forecasting of conversation derailment. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 7915–7919, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.
- Kennedy, B., Jin, X., Mostafazadeh Davani, A., Dehghani, M., and Ren, X. (2020a). Contextualizing hate speech classifiers with post-hoc explanation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5435–5442, Online. Association for Computational Linguistics.
- Kennedy, C. J., Bacon, G., Sahn, A., and von Vacano, C. (2020b). Constructing interval variables via faceted rasch measurement and multitask deep learning: a hate speech application. *arXiv preprint arXiv:2009.10277*.
- Khayrallah, H. and Koehn, P. (2018). On the impact of various types of noise on neural machine translation. In *Proceedings of the 2nd Workshop on Neural Machine Translation and Generation*, pages 74–83, Melbourne, Australia. Association for Computational Linguistics.

- Kiela, D., Bhooshan, S., Firooz, H., and Testuggine, D. (2019). Supervised multimodal bi-transformers for classifying images and text. In *Visually Grounded Interaction and Language (ViGIL), NeurIPS 2019 Workshop, Vancouver, Canada, December 13, 2019*.
- Kim, J., Derakhshan, A., and Harris, I. (2024). Robust safety classifier against jailbreaking attacks: Adversarial prompt shield. In Chung, Y.-L., Talat, Z., Nozza, D., Plaza-del Arco, F. M., Röttger, P., Mostafazadeh Davani, A., and Calabrese, A., editors, *Proceedings of the 8th Workshop on Online Abuse and Harms (WOAH 2024)*, pages 159–170, Mexico City, Mexico. Association for Computational Linguistics.
- Kim, W., Son, B., and Kim, I. (2021). Vilt: Vision-and-language transformer without convolution or region supervision. In *ICML*, pages 5583–5594.
- Kim, Y. (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.
- Kingma, D. P. and Ba, J. (2015a). Adam: A method for stochastic optimization. In Bengio, Y. and LeCun, Y., editors, *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.
- Kingma, D. P. and Ba, J. (2015b). Adam: A method for stochastic optimization. In Bengio, Y. and LeCun, Y., editors, *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.
- Kingma, D. P. and Welling, M. (2022). Auto-encoding variational bayes.
- Kocmi, T., Avramidis, E., Bawden, R., Bojar, O., Dvorkovich, A., Federmann, C., Fishel, M., Freitag, M., Gowda, T., Grundkiewicz, R., Haddow, B., Koehn, P., Marie, B., Monz, C., Morishita, M., Murray, K., Nagata, M., Nakazawa, T., Popel, M., Popović, M., and Shmatova, M. (2023). Findings of the 2023 conference on machine translation (WMT23): LLMs are here but not quite there yet. In Koehn, P., Haddow, B., Kocmi, T., and Monz, C., editors, *Proceedings of the Eighth Conference on Machine Translation*, pages 1–42, Singapore. Association for Computational Linguistics.

- Koehn, P., Hoang, H., Birch, A., Callison-Burch, C., Federico, M., Bertoldi, N., Cowan, B., Shen, W., Moran, C., Zens, R., Dyer, C., Bojar, O., Constantin, A., and Herbst, E. (2007). Moses: Open source toolkit for statistical machine translation. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics Companion Volume Proceedings of the Demo and Poster Sessions*, pages 177–180, Prague, Czech Republic. Association for Computational Linguistics.
- Kumar, A., Makhija, P., and Gupta, A. (2020). Noisy text data: Achilles’ heel of BERT. In *Proceedings of the Sixth Workshop on Noisy User-generated Text (W-NUT 2020)*, pages 16–21, Online. Association for Computational Linguistics.
- Kurita, K., Belova, A., and Anastasopoulos, A. (2020). Towards robust toxic content classification. arXiv:1912.06872.
- Lai, T. M., Hung Tran, Q., Bui, T., and Kihara, D. (2020). A simple but effective bert model for dialog state tracking on resource-limited systems. In *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 8034–8038.
- Lambert, C., Rajagopal, A., and Chandrasekharan, E. (2022). Conversational resilience: Quantifying and predicting conversational outcomes following adverse events. *Proceedings of the International AAAI Conference on Web and Social Media*, 16(1):548–559.
- Lan, Z., Chen, M., Goodman, S., Gimpel, K., Sharma, P., and Soricut, R. (2020). Albert: A lite bert for self-supervised learning of language representations. In *ICLR*. OpenReview.net.
- Lee, J., Cho, K., and Hofmann, T. (2017). Fully character-level neural machine translation without explicit segmentation. *Transactions of the Association for Computational Linguistics*, 5:365–378.
- Lewis, M., Liu, Y., Goyal, N., Ghazvininejad, M., Mohamed, A., Levy, O., Stoyanov, V., and Zettlemoyer, L. (2020). BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880, Online. Association for Computational Linguistics.

- Li, J., Ji, S., Du, T., Li, B., and Wang, T. (2019a). Textbugger: Generating adversarial text against real-world applications. In *26th Annual Network and Distributed System Security Symposium, NDSS 2019, San Diego, California, USA, February 24-27, 2019*. The Internet Society.
- Li, L., Ma, R., Guo, Q., Xue, X., and Qiu, X. (2020a). 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, Online. Association for Computational Linguistics.
- Li, L., Song, D., and Qiu, X. (2023). Text adversarial purification as defense against adversarial attacks. In Rogers, A., Boyd-Graber, J., and Okazaki, N., editors, *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 338–350, Toronto, Canada. Association for Computational Linguistics.
- Li, X., Michel, P., Anastasopoulos, A., Belinkov, Y., Durrani, N., Firat, O., Koehn, P., Neubig, G., Pino, J., and Sajjad, H. (2019b). Findings of the first shared task on machine translation robustness. In *Proceedings of the Fourth Conference on Machine Translation (Volume 2: Shared Task Papers, Day 1)*, pages 91–102, Florence, Italy. Association for Computational Linguistics.
- Li, Z., Fomicheva, M., and Specia, L. (2020b). Exploring model consensus to generate translation paraphrases. In *Proceedings of the Fourth Workshop on Neural Generation and Translation*, pages 161–168, Online. Association for Computational Linguistics.
- Li, Z., Rei, M., and Specia, L. (2021a). Visual cues and error correction for translation robustness. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 3153–3168, Punta Cana, Dominican Republic. Association for Computational Linguistics.
- Li, Z., Rei, M., and Specia, L. (2022). Multimodal conversation modelling for topic derailment detection. In Goldberg, Y., Kozareva, Z., and Zhang, Y., editors, *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 5115–5127, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

- Li, Z. and Specia, L. (2019). Improving neural machine translation robustness via data augmentation: Beyond back-translation. In *Proceedings of the 5th Workshop on Noisy User-generated Text (W-NUT 2019)*, pages 328–336, Hong Kong, China. Association for Computational Linguistics.
- Li, Z., Xu, J., Zeng, J., Li, L., Zheng, X., Zhang, Q., Chang, K.-W., and Hsieh, C.-J. (2021b). Searching for an effective defender: Benchmarking defense against adversarial word substitution. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 3137–3147, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.
- Li, Z., Zhou, H., Rei, M., and Specia, L. (2024). Diffusedef: Improved robustness to adversarial attacks via iterative denoising.
- Libovický, J. and Helcl, J. (2017). Attention strategies for multi-source sequence-to-sequence learning. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 196–202, Vancouver, Canada. Association for Computational Linguistics.
- Libovický, J., Helcl, J., and Mareček, D. (2018). Input combination strategies for multi-source transformer decoder. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 253–260, Brussels, Belgium. Association for Computational Linguistics.
- Liu, A., Yu, H., Hu, X., Li, S., Lin, L., Ma, F., Yang, Y., and Wen, L. (2022). Character-level white-box adversarial attacks against transformers via attachable subwords substitution. In Goldberg, Y., Kozareva, Z., and Zhang, Y., editors, *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 7664–7676, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.
- Liu, P., Qiu, X., and Huang, X. (2017). Adversarial multi-task learning for text classification. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1–10, Vancouver, Canada. Association for Computational Linguistics.

- Liu, Y., Deng, G., Li, Y., Wang, K., Wang, Z., Wang, X., Zhang, T., Liu, Y., Wang, H., Zheng, Y., and Liu, Y. (2024). Prompt injection attack against llm-integrated applications.
- Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L., and Stoyanov, V. (2019a). Roberta: A robustly optimized bert pretraining approach.
- Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L., and Stoyanov, V. (2019b). Roberta: A robustly optimized bert pretraining approach.
- Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L., and Stoyanov, V. (2019c). Roberta: A robustly optimized bert pretraining approach.
- Ma, E. (2019). Nlp augmentation. <https://github.com/makcedward/nlpaug>.
- Maas, A. L., Daly, R. E., Pham, P. T., Huang, D., Ng, A. Y., and Potts, C. (2011). Learning word vectors for sentiment analysis. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 142–150, Portland, Oregon, USA. Association for Computational Linguistics.
- Madry, A., Makelov, A., Schmidt, L., Tsipras, D., and Vladu, A. (2018). Towards deep learning models resistant to adversarial attacks. In *International Conference on Learning Representations*.
- Madry, A., Makelov, A., Schmidt, L., Tsipras, D., and Vladu, A. (2019). Towards deep learning models resistant to adversarial attacks.
- Malykh, V. (2019). Robust to noise models in natural language processing tasks. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics: Student Research Workshop*, pages 10–16, Florence, Italy. Association for Computational Linguistics.
- Mamta, M., Chigrupaatii, R., and Ekbal, A. (2024). BiasWipe: Mitigating unintended bias in text classifiers through model interpretability. In Al-Onaizan, Y., Bansal, M., and Chen, Y.-N., editors, *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 21059–21070, Miami, Florida, USA. Association for Computational Linguistics.

- Mathew, B., Saha, P., Yimam, S. M., Biemann, C., Goyal, P., and Mukherjee, A. (2021). Hatexplain: A benchmark dataset for explainable hate speech detection. *Proceedings of the AAAI Conference on Artificial Intelligence*, 35(17):14867–14875.
- McCoy, T., Pavlick, E., and Linzen, T. (2019). Right for the wrong reasons: Diagnosing syntactic heuristics in natural language inference. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3428–3448, Florence, Italy. Association for Computational Linguistics.
- Michel, P., Li, X., Neubig, G., and Pino, J. (2019). On evaluation of adversarial perturbations for sequence-to-sequence models. 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 3103–3114, Minneapolis, Minnesota. Association for Computational Linguistics.
- Michel, P. and Neubig, G. (2018). MTNT: A testbed for machine translation of noisy text. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 543–553, Brussels, Belgium. Association for Computational Linguistics.
- Miyato, T., Dai, A. M., and Goodfellow, I. (2016). Adversarial training methods for semi-supervised text classification.
- Moon, H. C., Joty, S., Zhao, R., Thakkar, M., and Xu, C. (2023). Randomized smoothing with masked inference for adversarially robust text classifications. In Rogers, A., Boyd-Graber, J., and Okazaki, N., editors, *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5145–5165, Toronto, Canada. Association for Computational Linguistics.
- Morris, J., Lifland, E., Lanchantin, J., Ji, Y., and Qi, Y. (2020a). Reevaluating adversarial examples in natural language. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 3829–3839, Online. Association for Computational Linguistics.
- Morris, J., Lifland, E., Yoo, J. Y., Grigsby, J., Jin, D., and Qi, Y. (2020b). TextAttack: A framework for adversarial attacks, data augmentation, and adversarial training in NLP. In

- Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 119–126, Online. Association for Computational Linguistics.
- Nangia, N., Vania, C., Bhalerao, R., and Bowman, S. R. (2020). CrowS-pairs: A challenge dataset for measuring social biases in masked language models. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1953–1967, Online. Association for Computational Linguistics.
- Náplava, J., Popel, M., Straka, M., and Straková, J. (2021). Understanding model robustness to user-generated noisy texts. In *Proceedings of the Seventh Workshop on Noisy User-generated Text (W-NUT 2021)*, pages 340–350, Online. Association for Computational Linguistics.
- Neerudu, P. K. R., Oota, S., Marreddy, M., Kagita, V., and Gupta, M. (2023). On robustness of finetuned transformer-based NLP models. In Bouamor, H., Pino, J., and Bali, K., editors, *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 7180–7195, Singapore. Association for Computational Linguistics.
- Ng, H. T., Wu, S. M., Briscoe, T., Hadiwinoto, C., Susanto, R. H., and Bryant, C. (2014). The CoNLL-2014 shared task on grammatical error correction. In *Proceedings of the Eighteenth Conference on Computational Natural Language Learning: Shared Task*, pages 1–14, Baltimore, Maryland. Association for Computational Linguistics.
- Nguyen, M. and Chen, N. (2023). FiRo: Finite-context indexing of restricted output space for NLP models facing noisy input. In Park, J. C., Arase, Y., Hu, B., Lu, W., Wijaya, D., Purwarianti, A., and Krisnadhi, A. A., editors, *Proceedings of the 13th International Joint Conference on Natural Language Processing and the 3rd Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 834–845, Nusa Dua, Bali. Association for Computational Linguistics.
- Nguyen, T. Q. and Salazar, J. (2019). Transformers without tears: Improving the normalization of self-attention. *arXiv preprint arXiv:1910.05895*.
- Nguyen Minh, D. and Luu, A. T. (2022). Textual manifold-based defense against natural language adversarial examples. In Goldberg, Y., Kozareva, Z., and Zhang, Y., editors, *Pro-*

- ceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 6612–6625, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.
- Nie, W., Guo, B., Huang, Y., Xiao, C., Vahdat, A., and Anandkumar, A. (2022). Diffusion models for adversarial purification. In *International Conference on Machine Learning (ICML)*.
- Nozza, D., Volpetti, C., and Fersini, E. (2019). Unintended bias in misogyny detection. In *IEEE/WIC/ACM International Conference on Web Intelligence, WI '19*, page 149–155, New York, NY, USA. Association for Computing Machinery.
- OpenAI, Achiam, J., Adler, S., Agarwal, S., Ahmad, L., Akkaya, I., Aleman, F. L., Almeida, D., Altenschmidt, J., Altman, S., Anadkat, S., Avila, R., Babuschkin, I., Balaji, S., Balcom, V., Baltescu, P., Bao, H., Bavarian, M., Belgum, J., Bello, I., Berdine, J., Bernadett-Shapiro, G., Berner, C., Bogdonoff, L., Boiko, O., Boyd, M., Brakman, A.-L., Brockman, G., Brooks, T., Brundage, M., Button, K., Cai, T., Campbell, R., Cann, A., Carey, B., Carlson, C., Carmichael, R., Chan, B., Chang, C., Chantzis, F., Chen, D., Chen, S., Chen, R., Chen, J., Chen, M., Chess, B., Cho, C., Chu, C., Chung, H. W., Cummings, D., Currier, J., Dai, Y., Decareaux, C., Degry, T., Deutsch, N., Deville, D., Dhar, A., Dohan, D., Dowling, S., Dunning, S., Ecoffet, A., Eleti, A., Eloundou, T., Farhi, D., Fedus, L., Felix, N., Fishman, S. P., Forte, J., Fulford, I., Gao, L., Georges, E., Gibson, C., Goel, V., Gogineni, T., Goh, G., Gontijo-Lopes, R., Gordon, J., Grafstein, M., Gray, S., Greene, R., Gross, J., Gu, S. S., Guo, Y., Hallacy, C., Han, J., Harris, J., He, Y., Heaton, M., Heidecke, J., Hesse, C., Hickey, A., Hickey, W., Hoeschele, P., Houghton, B., Hsu, K., Hu, S., Hu, X., Huizinga, J., Jain, S., Jain, S., Jang, J., Jiang, A., Jiang, R., Jin, H., Jin, D., Jomoto, S., Jonn, B., Jun, H., Kaftan, T., Łukasz Kaiser, Kamali, A., Kanitscheider, I., Keskar, N. S., Khan, T., Kilpatrick, L., Kim, J. W., Kim, C., Kim, Y., Kirchner, J. H., Kiros, J., Knight, M., Kokotajlo, D., Łukasz Kondraciuk, Kondrich, A., Konstantinidis, A., Kosic, K., Krueger, G., Kuo, V., Lampe, M., Lan, I., Lee, T., Leike, J., Leung, J., Levy, D., Li, C. M., Lim, R., Lin, M., Lin, S., Litwin, M., Lopez, T., Lowe, R., Lue, P., Makanju, A., Malfacini, K., Manning, S., Markov, T., Markovski, Y., Martin, B., Mayer, K., Mayne, A., McGrew, B., McKinney, S. M., McLeavey,

C., McMillan, P., McNeil, J., Medina, D., Mehta, A., Menick, J., Metz, L., Mishchenko, A., Mishkin, P., Monaco, V., Morikawa, E., Mossing, D., Mu, T., Murati, M., Murk, O., Mély, D., Nair, A., Nakano, R., Nayak, R., Neelakantan, A., Ngo, R., Noh, H., Ouyang, L., O’Keefe, C., Pachocki, J., Paino, A., Palermo, J., Pantuliano, A., Parascandolo, G., Parish, J., Parparita, E., Passos, A., Pavlov, M., Peng, A., Perelman, A., de Avila Belbute Peres, F., Petrov, M., de Oliveira Pinto, H. P., Michael, Pokorny, Pokrass, M., Pong, V. H., Powell, T., Power, A., Power, B., Proehl, E., Puri, R., Radford, A., Rae, J., Ramesh, A., Raymond, C., Real, F., Rimbach, K., Ross, C., Rotsted, B., Roussez, H., Ryder, N., Saltarelli, M., Sanders, T., Santurkar, S., Sastry, G., Schmidt, H., Schnurr, D., Schulman, J., Selsam, D., Sheppard, K., Sherbakov, T., Shieh, J., Shoker, S., Shyam, P., Sidor, S., Sigler, E., Simens, M., Sitkin, J., Slama, K., Sohl, I., Sokolowsky, B., Song, Y., Staudacher, N., Such, F. P., Summers, N., Sutskever, I., Tang, J., Tezak, N., Thompson, M. B., Tillet, P., Tootoonchian, A., Tseng, E., Tuggle, P., Turley, N., Tworek, J., Uribe, J. F. C., Vallone, A., Vijayvergiya, A., Voss, C., Wainwright, C., Wang, J. J., Wang, A., Wang, B., Ward, J., Wei, J., Weinmann, C., Welihinda, A., Welinder, P., Weng, J., Weng, L., Wiethoff, M., Willner, D., Winter, C., Wolrich, S., Wong, H., Workman, L., Wu, S., Wu, J., Wu, M., Xiao, K., Xu, T., Yoo, S., Yu, K., Yuan, Q., Zaremba, W., Zellers, R., Zhang, C., Zhang, M., Zhao, S., Zheng, T., Zhuang, J., Zhuk, W., and Zoph, B. (2024). Gpt-4 technical report.

Ousidhoum, N., Lin, Z., Zhang, H., Song, Y., and Yeung, D.-Y. (2019). Multilingual and multi-aspect hate speech analysis. 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)*, pages 4675–4684, Hong Kong, China. Association for Computational Linguistics.

Ousidhoum, N., Zhao, X., Fang, T., Song, Y., and Yeung, D.-Y. (2021). Probing toxic content in large pre-trained language 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 4262–4274, Online. Association for Computational Linguistics.

- Pappagari, R., Zelasko, P., Villalba, J., Carmiel, Y., and Dehak, N. (2019). Hierarchical transformers for long document classification. In *2019 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, pages 838–844.
- Park, C. Y., Mendelsohn, J., Radhakrishnan, K., Jain, K., Kanakagiri, T., Jurgens, D., and Tsvetkov, Y. (2021). Detecting community sensitive norm violations in online conversations. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 3386–3397, Punta Cana, Dominican Republic. Association for Computational Linguistics.
- Park, J. H., Shin, J., and Fung, P. (2018). Reducing gender bias in abusive language detection. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2799–2804, Brussels, Belgium. Association for Computational Linguistics.
- Passban, P., Saladi, P. S. M., and Liu, Q. (2020). Revisiting robust neural machine translation: A transformer case study.
- Peng, M., Zhang, Q., Jiang, Y.-g., and Huang, X. (2018). Cross-domain sentiment classification with target domain specific information. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2505–2513, Melbourne, Australia. Association for Computational Linguistics.
- Peskov, D., Barrow, J., Rodriguez, P., Neubig, G., and Boyd-Graber, J. (2019). Mitigating noisy inputs for question answering. In *Conference of the International Speech Communication Association*.
- Plaza-del Arco, F. M., Halat, S., Padó, S., and Klinger, R. (2021). Multi-task learning with sentiment, emotion, and target detection to recognize hate speech and offensive language. *arXiv preprint arXiv:2109.10255*.
- Pruthi, D., Dhingra, B., and Lipton, Z. C. (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.

- Qi, X., Zeng, Y., Xie, T., Chen, P.-Y., Jia, R., Mittal, P., and Henderson, P. (2023). Fine-tuning aligned language models compromises safety, even when users do not intend to!
- Qian, J., ElSherief, M., Belding, E., and Wang, W. Y. (2018). Hierarchical CVAE for fine-grained hate speech classification. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3550–3559, Brussels, Belgium. Association for Computational Linguistics.
- Radford, A. and Narasimhan, K. (2018). Improving language understanding by generative pre-training.
- Rafailov, R., Sharma, A., Mitchell, E., Ermon, S., Manning, C. D., and Finn, C. (2024). Direct preference optimization: Your language model is secretly a reward model.
- Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., Zhou, Y., Li, W., and Liu, P. J. (2020). Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140):1–67.
- Rajpurkar, P., Jia, R., and Liang, P. (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.
- Rajpurkar, P., Zhang, J., Lopyrev, K., and Liang, P. (2016). SQuAD: 100,000+ questions for machine comprehension of text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2383–2392, Austin, Texas. Association for Computational Linguistics.
- Ramesh, A., Pavlov, M., Goh, G., Gray, S., Voss, C., Radford, A., Chen, M., and Sutskever, I. (2021). Zero-shot text-to-image generation.
- Ren, S., Deng, Y., He, K., and Che, W. (2019). Generating natural language adversarial examples through probability weighted word saliency. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1085–1097, Florence, Italy. Association for Computational Linguistics.

- Ren, S., He, K., Girshick, R., and Sun, J. (2015). Faster r-cnn: Towards real-time object detection with region proposal networks. In Cortes, C., Lawrence, N., Lee, D., Sugiyama, M., and Garnett, R., editors, *Advances in Neural Information Processing Systems*, volume 28. Curran Associates, Inc.
- Ribeiro, M. T., Wu, T., Guestrin, C., and Singh, S. (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.
- Rombach, R., Blattmann, A., Lorenz, D., Esser, P., and Ommer, B. (2021). High-resolution image synthesis with latent diffusion models.
- Rosales Núñez, J. C., Wisniewski, G., and Seddah, D. (2021). Noisy UGC translation at the character level: Revisiting open-vocabulary capabilities and robustness of char-based models. In *Proceedings of the Seventh Workshop on Noisy User-generated Text (W-NUT 2021)*, pages 199–211, Online. Association for Computational Linguistics.
- Röttger, P., Vidgen, B., Nguyen, D., Waseem, Z., Margetts, H., and Pierrehumbert, J. (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.
- Salesky, E., Etter, D., and Post, M. (2021). Robust open-vocabulary translation from visual text representations.
- Santra, B., Anusha, P., and Goyal, P. (2021). Hierarchical transformer for task oriented dialog systems. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 5649–5658, Online. Association for Computational Linguistics.
- Sap, M., Card, D., Gabriel, S., Choi, Y., and Smith, N. A. (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.

Schick, T., Udupa, S., and Schütze, H. (2021). Self-diagnosis and self-debiasing: A proposal for reducing corpus-based bias in nlp. *Transactions of the Association for Computational Linguistics*, 9:1408–1424.

Schmidt, A. and Wiegand, M. (2017). A survey on hate speech detection using natural language processing. In *Proceedings of the Fifth International Workshop on Natural Language Processing for Social Media*, pages 1–10, Valencia, Spain. Association for Computational Linguistics.

Sengupta, S., Krone, J., and Mansour, S. (2021). On the robustness of intent classification and slot labeling in goal-oriented dialog systems to real-world noise. In *Proceedings of the 3rd Workshop on Natural Language Processing for Conversational AI*, pages 68–79, Online. Association for Computational Linguistics.

Serban, I., Sordoni, A., Lowe, R., Charlin, L., Pineau, J., Courville, A., and Bengio, Y. (2017). A hierarchical latent variable encoder-decoder model for generating dialogues. *Proceedings of the AAAI Conference on Artificial Intelligence*, 31(1).

Shepherd, T., Harvey, A., Jordan, T., Srauy, S., and Miltner, K. (2015). Histories of hating. *Social Media + Society*, 1(2):2056305115603997.

Shi, Z., Zhang, H., Chang, K.-W., Huang, M., and Hsieh, C.-J. (2020). Robustness verification for transformers.

Shriberg, E. (1994). *Preliminaries to a Theory of Speech Disfluencies*. PhD thesis, University of California, Berkeley, CA.

Shuster, K., Xu, J., Komeili, M., Ju, D., Smith, E. M., Roller, S., Ung, M., Chen, M., Arora, K., Lane, J., Behrooz, M., Ngan, W., Poff, S., Goyal, N., Szlam, A., Boureau, Y.-L., Kambadur, M., and Weston, J. (2022). Blenderbot 3: a deployed conversational agent that continually learns to responsibly engage.

- Si, C., Zhang, Z., Qi, F., Liu, Z., Wang, Y., Liu, Q., and Sun, M. (2021). Better robustness by more coverage: Adversarial and mixup data augmentation for robust finetuning. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 1569–1576, Online. Association for Computational Linguistics.
- Singh, A., Singh, N., and Vatsal, S. (2024). Robustness of llms to perturbations in text.
- Sohl-Dickstein, J., Weiss, E., Maheswaranathan, N., and Ganguli, S. (2015). Deep unsupervised learning using nonequilibrium thermodynamics. In Bach, F. and Blei, D., editors, *Proceedings of the 32nd International Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*, pages 2256–2265, Lille, France. PMLR.
- Song, J., Meng, C., and Ermon, S. (2021). Denoising diffusion implicit models. In *International Conference on Learning Representations*.
- Song, K., Tan, X., Qin, T., Lu, J., and Liu, T.-Y. (2019). Mass: Masked sequence to sequence pre-training for language generation. In *International Conference on Machine Learning*, pages 5926–5936.
- Sordoni, A., Bengio, Y., Vahabi, H., Lioma, C., Grue Simonsen, J., and Nie, J.-Y. (2015). A hierarchical recurrent encoder-decoder for generative context-aware query suggestion. In *Proceedings of the 24th ACM International on Conference on Information and Knowledge Management, CIKM '15*, page 553–562, New York, NY, USA. Association for Computing Machinery.
- Specia, L., Blain, F., Fomicheva, M., Zerva, C., Li, Z., Chaudhary, V., and Martins, A. F. T. (2021). Findings of the WMT 2021 shared task on quality estimation. In *Proceedings of the Sixth Conference on Machine Translation*, pages 684–725, Online. Association for Computational Linguistics.
- Specia, L., Fomicheva, M., Caglayan, O., and Li, Z. (2023). Multi-task neural network for toxicity detection.
- Specia, L., Frank, S., Sima'an, K., and Elliott, D. (2016a). A shared task on multimodal machine translation and crosslingual image description. In *Proceedings of the First Conference*

- on Machine Translation: Volume 2, Shared Task Papers*, pages 543–553, Berlin, Germany. Association for Computational Linguistics.
- Specia, L., Frank, S., Sima'an, K., and Elliott, D. (2016b). A shared task on multimodal machine translation and crosslingual image description. In *First Conference on Machine Translation, Volume 2: Shared Task Papers*, WMT, pages 540–550, Berlin, Germany.
- Specia, L., Li, Z., Pino, J., Chaudhary, V., Guzmán, F., Neubig, G., Durrani, N., Belinkov, Y., Koehn, P., Sajjad, H., Michel, P., and Li, X. (2020a). Findings of the WMT 2020 shared task on machine translation robustness. In *Proceedings of the Fifth Conference on Machine Translation*, pages 76–91, Online. Association for Computational Linguistics.
- Specia, L., Li, Z., Pino, J., Chaudhary, V., Guzmán, F., Neubig, G., Durrani, N., Belinkov, Y., Koehn, P., Sajjad, H., Michel, P., and Li, X. (2020b). Findings of the wmt 2020 shared task on machine translation robustness. In *Proceedings of the Fifth Conference on Machine Translation*, pages 76–91, Online. Association for Computational Linguistics.
- Srivastava, A. and Chiang, D. (2023). BERTwch: Extending BERT’s capabilities to model dialectal and noisy text. In Bouamor, H., Pino, J., and Bali, K., editors, *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 15510–15521, Singapore. Association for Computational Linguistics.
- Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., and Salakhutdinov, R. (2014). Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15(56):1929–1958.
- Sulubacak, U., Caglayan, O., Grönroos, S.-A., Rouhe, A., Elliott, D., Specia, L., and Tiedemann, J. (2020). Multimodal machine translation through visuals and speech. *Machine Translation*, 34(2):97–147.
- Sun, Y. and Jiang, H. (2019). Contextual text denoising with masked language model. In *Proceedings of the 5th Workshop on Noisy User-generated Text (W-NUT 2019)*, pages 286–290, Hong Kong, China. Association for Computational Linguistics.

- Sundararajan, M., Taly, A., and Yan, Q. (2017). Axiomatic attribution for deep networks. In *Proceedings of the 34th International Conference on Machine Learning - Volume 70*, ICML'17, page 3319–3328. JMLR.org.
- Sutskever, I., Vinyals, O., and Le, Q. V. (2014). Sequence to sequence learning with neural networks. In *Proceedings of the 27th International Conference on Neural Information Processing Systems - Volume 2*, NIPS'14, page 3104–3112, Cambridge, MA, USA. MIT Press.
- Szegedy, C., Zaremba, W., Sutskever, I., Bruna, J., Erhan, D., Goodfellow, I. J., and Fergus, R. (2013). Intriguing properties of neural networks. *CoRR*, abs/1312.6199.
- Tamirisa, R., Bharathi, B., Phan, L., Zhou, A., Gatti, A., Suresh, T., Lin, M., Wang, J., Wang, R., Arel, R., Zou, A., Song, D., Li, B., Hendrycks, D., and Mazeika, M. (2024). Tamper-resistant safeguards for open-weight llms.
- Thoppilan, R., Freitas, D. D., Hall, J., Shazeer, N., Kulshreshtha, A., Cheng, H.-T., Jin, A., Bos, T., Baker, L., Du, Y., Li, Y., Lee, H., Zheng, H. S., Ghafouri, A., Menegali, M., Huang, Y., Krikun, M., Lepikhin, D., Qin, J., Chen, D., Xu, Y., Chen, Z., Roberts, A., Bosma, M., Zhao, V., Zhou, Y., Chang, C.-C., Krivokon, I., Rusch, W., Pickett, M., Srinivasan, P., Man, L., Meier-Hellstern, K., Morris, M. R., Doshi, T., Santos, R. D., Duke, T., Soraker, J., Zevenbergen, B., Prabhakaran, V., Diaz, M., Hutchinson, B., Olson, K., Molina, A., Hoffman-John, E., Lee, J., Aroyo, L., Rajakumar, R., Butryna, A., Lamm, M., Kuzmina, V., Fenton, J., Cohen, A., Bernstein, R., Kurzweil, R., Agueria-Arcas, B., Cui, C., Croak, M., Chi, E., and Le, Q. (2022). Lamda: Language models for dialog applications.
- Touvron, H., Lavril, T., Izacard, G., Martinet, X., Lachaux, M.-A., Lacroix, T., Rozière, B., Goyal, N., Hambro, E., Azhar, F., Rodriguez, A., Joulin, A., Grave, E., and Lample, G. (2023). Llama: Open and efficient foundation language models.
- Turbal, B., Mazur, A., Zhao, J., and Pechenizkiy, M. (2024). On adversarial robustness of language models in transfer learning. In *Workshop on Socially Responsible Language Modelling Research*.

- Tzeng, E., Hoffman, J., Darrell, T., and Saenko, K. (2015). Simultaneous deep transfer across domains and tasks. In *2015 IEEE International Conference on Computer Vision (ICCV)*, pages 4068–4076, Los Alamitos, CA, USA. IEEE Computer Society.
- Vaibhav, V., Singh, S., Stewart, C., and Neubig, G. (2019). Improving robustness of machine translation with synthetic noise. 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 1916–1920, Minneapolis, Minnesota. Association for Computational Linguistics.
- Vaidya, A., Mai, F., and Ning, Y. (2020). Empirical analysis of multi-task learning for reducing identity bias in toxic comment detection. *Proceedings of the International AAAI Conference on Web and Social Media*, 14(1):683–693.
- van der Maaten, L. and Hinton, G. (2008). Visualizing data using t-sne. *Journal of Machine Learning Research*, 9(86):2579–2605.
- Vassilev, A., Oprea, A., Fordyce, A., and Andersen, H. (2024). Adversarial machine learning: A taxonomy and terminology of attacks and mitigations. *NIST Artificial Intelligence (AI) Report*.
- Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., and Polosukhin, I. (2017). Attention is all you need. In *Proceedings of the 31st International Conference on Neural Information Processing Systems, NIPS’17*, page 6000–6010, Red Hook, NY, USA. Curran Associates Inc.
- Vidgen, B., Nguyen, D., Margetts, H., Rossini, P., and Tromble, R. (2021). Introducing CAD: the contextual abuse dataset. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2289–2303, Online. Association for Computational Linguistics.
- Vig, J., Gehrmann, S., Belinkov, Y., Qian, S., Nevo, D., Singer, Y., and Shieber, S. (2020). Investigating gender bias in language models using causal mediation analysis. In Larochelle, H.,

- Ranzato, M., Hadsell, R., Balcan, M., and Lin, H., editors, *Advances in Neural Information Processing Systems*, volume 33, pages 12388–12401. Curran Associates, Inc.
- Wang, A., Pruksachatkun, Y., Nangia, N., Singh, A., Michael, J., Hill, F., Levy, O., and Bowman, S. (2019). Superglue: A stickier benchmark for general-purpose language understanding systems. In Wallach, H., Larochelle, H., Beygelzimer, A., d'Alché-Buc, F., Fox, E., and Garnett, R., editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc.
- Wang, A., Singh, A., Michael, J., Hill, F., Levy, O., and Bowman, S. (2018). GLUE: A multi-task benchmark and analysis platform for natural language understanding. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 353–355, Brussels, Belgium. Association for Computational Linguistics.
- Wang, B., Wang, S., Cheng, Y., Gan, Z., Jia, R., Li, B., and Liu, J. (2021). Infobert: Improving robustness of language models from an information theoretic perspective. In *International Conference on Learning Representations*.
- Wang, B., Wei, C., Liu, Z., Lin, G., and Chen, N. F. (2024). Resilience of large language models for noisy instructions. In Al-Onaizan, Y., Bansal, M., and Chen, Y.-N., editors, *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 11939–11950, Miami, Florida, USA. Association for Computational Linguistics.
- Wang, B., Xu, C., Liu, X., Cheng, Y., and Li, B. (2022a). SemAttack: Natural textual attacks via different semantic spaces. In *Findings of the Association for Computational Linguistics: NAACL 2022*, pages 176–205, Seattle, United States. Association for Computational Linguistics.
- Wang, X., Wang, H., and Yang, D. (2022b). Measure and improve robustness in NLP models: A survey. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4569–4586, Seattle, United States. Association for Computational Linguistics.

- Wang, Z., Liu, Z., Zheng, X., Su, Q., and Wang, J. (2023). RMLM: A flexible defense framework for proactively mitigating word-level adversarial attacks. In Rogers, A., Boyd-Graber, J., and Okazaki, N., editors, *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2757–2774, Toronto, Canada. Association for Computational Linguistics.
- Wang, Z., Qinami, K., Karakozis, I., Genova, K., Nair, P., Hata, K., and Russakovsky, O. (2020). Towards fairness in visual recognition: Effective strategies for bias mitigation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.
- Wu, Y. and Guo, Y. (2020). Dual adversarial co-learning for multi-domain text classification. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(04):6438–6445.
- Wu, Y., Inkpen, D., and El-Roby, A. (2021). Conditional adversarial networks for multi-domain text classification. In *Proceedings of the Second Workshop on Domain Adaptation for NLP*, pages 16–27, Kyiv, Ukraine. Association for Computational Linguistics.
- Wu, Z., Caglayan, O., Ive, J., Wang, J., and Specia, L. (2019a). Transformer-based cascaded multimodal speech translation. In *Proceedings of the 16th International Conference on Spoken Language Translation*, Hong Kong. Association for Computational Linguistics.
- Wu, Z., Ive, J., Wang, J., Madhyastha, P., and Specia, L. (2019b). Predicting actions to help predict translations. *CoRR*, abs/1908.01665.
- Xi, Z., Zheng, R., Gui, T., Zhang, Q., and Huang, X. (2022). Efficient adversarial training with robust early-bird tickets. In Goldberg, Y., Kozareva, Z., and Zhang, Y., editors, *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 8318–8331, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.
- Xie, Y., Yi, J., Shao, J., Curl, J., Lyu, L., Chen, Q., Xie, X., and Wu, F. (2023). Defending chatgpt against jailbreak attack via self-reminders. *Nature Machine Intelligence*, 5:1486–1496.
- Xu, W., Aw, A. T., Ding, Y., Wu, K., and Joty, S. (2021a). Addressing the vulnerability of nmt in input perturbations.

- Xu, Y., Zhong, X., Jimeno Yepes, A., and Lau, J. H. (2021b). Grey-box adversarial attack and defence for sentiment classification. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4078–4087, Online. Association for Computational Linguistics.
- Xu, Z., Liu, Y., Deng, G., Li, Y., and Picek, S. (2024). A comprehensive study of jailbreak attack versus defense for large language models. In Ku, L.-W., Martins, A., and Srikumar, V., editors, *Findings of the Association for Computational Linguistics: ACL 2024*, pages 7432–7449, Bangkok, Thailand. Association for Computational Linguistics.
- Xue, L., Barua, A., Constant, N., Al-Rfou, R., Narang, S., Kale, M., Roberts, A., and Raffel, C. (2022). ByT5: Towards a token-free future with pre-trained byte-to-byte models. *Transactions of the Association for Computational Linguistics*, 10:291–306.
- Yang, P., Tan, Q., Ye, J., Tong, H., and He, J. (2019a). Deep multi-task learning with adversarial-and-cooperative nets. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI-19*, pages 4078–4084. International Joint Conferences on Artificial Intelligence Organization.
- Yang, Z., Dai, Z., Yang, Y., Carbonell, J., Salakhutdinov, R. R., and Le, Q. V. (2019b). Xlnet: Generalized autoregressive pretraining for language understanding. In Wallach, H., Larochelle, H., Beygelzimer, A., d'Alché-Buc, F., Fox, E., and Garnett, R., editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc.
- Yang, Z., Dai, Z., Yang, Y., Carbonell, J., Salakhutdinov, R. R., and Le, Q. V. (2019c). Xlnet: Generalized autoregressive pretraining for language understanding. In Wallach, H., Larochelle, H., Beygelzimer, A., d'Alché-Buc, F., Fox, E., and Garnett, R., editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc.
- Yang, Z., Yang, D., Dyer, C., He, X., Smola, A., and Hovy, E. (2016). Hierarchical attention networks for document classification. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Lan-*

- guage Technologies*, pages 1480–1489, San Diego, California. Association for Computational Linguistics.
- Yao, S. and Wan, X. (2020). Multimodal transformer for multimodal machine translation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4346–4350, Online. Association for Computational Linguistics.
- Ye, M., Gong, C., and Liu, Q. (2020). SAFER: A structure-free approach for certified robustness to adversarial word substitutions. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 3465–3475, Online. Association for Computational Linguistics.
- Ye, W., Zheng, G., Cao, X., Ma, Y., and Zhang, A. (2024). Spurious correlations in machine learning: A survey.
- Yoo, J. Y. and Qi, Y. (2021). Towards improving adversarial training of NLP models. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 945–956, Punta Cana, Dominican Republic. Association for Computational Linguistics.
- Yuan, L., Zhang, Y., Chen, Y., and Wei, W. (2023). Bridge the gap between CV and NLP! a gradient-based textual adversarial attack framework. In Rogers, A., Boyd-Graber, J., and Okazaki, N., editors, *Findings of the Association for Computational Linguistics: ACL 2023*, pages 7132–7146, Toronto, Canada. Association for Computational Linguistics.
- Yuan, S., Yuan, W., Yin, H., and He, T. (2024). Roic-dm: Robust text inference and classification via diffusion model.
- Yuan, Z. and Briscoe, T. (2016). Grammatical error correction using neural machine translation. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 380–386, San Diego, California. Association for Computational Linguistics.
- Zang, Y., Qi, F., Yang, C., Liu, Z., Zhang, M., Liu, Q., and Sun, M. (2020). Word-level textual adversarial attacking as combinatorial optimization. In *Proceedings of the 58th Annual Meet-*

- ing of the Association for Computational Linguistics*, pages 6066–6080, Online. Association for Computational Linguistics.
- Zeng, J., Xu, J., Zheng, X., and Huang, X. (2023). Certified robustness to text adversarial attacks by randomized [MASK]. *Computational Linguistics*, 49(2):395–427.
- Zhang, H., Lan, Y., Pang, L., Guo, J., and Cheng, X. (2019a). ReCoSa: Detecting the relevant contexts with self-attention for multi-turn dialogue generation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3721–3730, Florence, Italy. Association for Computational Linguistics.
- Zhang, J., Chang, J., Danescu-Niculescu-Mizil, C., Dixon, L., Hua, Y., Taraborelli, D., and Thain, N. (2018a). Conversations gone awry: Detecting early signs of conversational failure. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1350–1361, Melbourne, Australia. Association for Computational Linguistics.
- Zhang, J., Danescu-Niculescu-Mizil, C., Sauper, C., and Taylor, S. J. (2018b). Characterizing online public discussions through patterns of participant interactions. *Proc. ACM Hum.-Comput. Interact.*, 2(CSCW).
- Zhang, X., Wei, F., and Zhou, M. (2019b). HIBERT: Document level pre-training of hierarchical bidirectional transformers for document summarization. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5059–5069, Florence, Italy. Association for Computational Linguistics.
- Zhang, X., Zhao, J., and LeCun, Y. (2015). Character-level convolutional networks for text classification. In *Proceedings of the 28th International Conference on Neural Information Processing Systems - Volume 1*, NIPS’15, page 649–657, Cambridge, MA, USA. MIT Press.
- Zhang, Y., Bartley, T. M., Graterol-Fuenmayor, M., Lavrukhin, V., Bakhturina, E., and Ginsburg, B. (2024). A chat about boring problems: Studying gpt-based text normalization.
- Zhang, Y., Sun, S., Galley, M., Chen, Y.-C., Brockett, C., Gao, X., Gao, J., Liu, J., and Dolan, B. (2020). DIALOGPT : Large-scale generative pre-training for conversational response

- generation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 270–278, Online. Association for Computational Linguistics.
- Zhao, J., Wang, T., Yatskar, M., Ordonez, V., and Chang, K.-W. (2018). Gender bias in coreference resolution: Evaluation and debiasing methods. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 15–20, New Orleans, Louisiana. Association for Computational Linguistics.
- Zhao, Y., Yan, L., Sun, W., Xing, G., Wang, S., Meng, C., Cheng, Z., Ren, Z., and Yin, D. (2024). Improving the robustness of large language models via consistency alignment.
- Zhong, M., Liu, Y., Xu, Y., Zhu, C., and Zeng, M. (2021). Dialoglm: Pre-trained model for long dialogue understanding and summarization.
- Zhong, M., Liu, Y., Xu, Y., Zhu, C., and Zeng, M. (2022). Dialoglm: Pre-trained model for long dialogue understanding and summarization. *Proceedings of the AAAI Conference on Artificial Intelligence*, 36(10):11765–11773.
- Zhou, X., Sap, M., Swayamdipta, S., Choi, Y., and Smith, N. (2021a). 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.
- Zhou, X., Yong, Y., Fan, X., Ren, G., Song, Y., Diao, Y., Yang, L., and Lin, H. (2021b). Hate speech detection based on sentiment knowledge sharing. 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 7158–7166, Online. Association for Computational Linguistics.
- Zhou, Y., Jiang, J.-Y., Chang, K.-W., and Wang, W. (2019). Learning to discriminate perturbations for blocking adversarial attacks in text classification. 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), pages 4904–4913, Hong Kong, China. Association for Computational Linguistics.

Zhou, Y., Zheng, X., Hsieh, C.-J., Chang, K.-W., and Huang, X. (2021c). Defense against synonym substitution-based adversarial attacks via Dirichlet neighborhood ensemble. 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 5482–5492, Online. Association for Computational Linguistics.

Zhu, C., Cheng, Y., Gan, Z., Sun, S., Goldstein, T., and Liu, J. (2020). Freelb: Enhanced adversarial training for natural language understanding. In *International Conference on Learning Representations*.

Zhu, Y., Kiros, R., Zemel, R., Salakhutdinov, R., Urtasun, R., Torralba, A., and Fidler, S. (2015). Aligning books and movies: Towards story-like visual explanations by watching movies and reading books.