EDOARDO MOSCA

EXPLAINABLE AI FOR THE HUMAN-CENTRIC DEVELOPMENT OF NLP MODELS

WORKING TOWARDS MORE INTERPRETABLE, ROBUST, AND CONTROLLABLE MODELS

Technische Universität München

TUM School of Computation, Information, and Technology

EXPLAINABLE AI FOR THE HUMAN-CENTRIC DEVELOPMENT OF NLP MODELS

EDOARDO MOSCA

Vollständiger Abdruck der von der TUM School of Computation, Information and Technology (CIT) der Technischen Universität München zur Erlangung des akademischen Grades eines

Doktors der Naturwissenschaften (Dr. rer. nat.)

genehmigten Dissertation.

Vorsitz:		Prof. Dr-Ing. Jörg Ott
Prüfende der Dissertation:	1.	apl. Prof. Dr. Georg Groh
	2.	Associate Prof. Dr. Alan Said
	3.	Prof. Dr. Diana Rieger

Die Dissertation wurde am 19.09.2023 bei der Technischen Universität München eingereicht und durch die TUM School of Computation, Information, and Technology am 15.07.2024 angenommen.

This document was typeset using LaTeX and the `classicthesis` template developed by André Miede and Ivo Pletikosić.

Explainable AI is critical for building trust in and accountability for AI systems. Without transparency into the decision-making process of AI, it is impossible to ensure that the technology is being used ethically and for the benefit of society.

— ChatGPT, OpenAI

February 2023

(Prompt: *Write a short quote about why XAI is important.*)

ABSTRACT

Larger and more complex models have consistently raised the performance bar in most *Natural Language Processing* (NLP) applications, exhibiting a growing presence in society and decision-making processes. However, their black-box nature raises significant concerns regarding their trustworthiness, as their scale and complexity hinder our ability to understand and control them. In response to this challenge, the development of human-centric NLP models has emerged as a priority, with European regulations defining key requirements to ensure that deployed systems align with human values and ultimately benefit society. This dissertation presents eight studies investigating the usage of model explanations to address the three key requirements of (1) *interpretability*, (2) *robustness*, and (3) *human oversight*. Specifically, we contribute by reviewing existing explainability methods, assessing their applicability to NLP, and developing approaches tailored to (multi-modal) NLP inputs in context-dependent applications. We show that model explanations carry strong signals enabling the explicit and model-agnostic detection of adversarial text attacks. Finally, we propose a human-model interaction platform, enabling annotators to influence and control deployed models by editing model explanations and thus providing human feedback. The insights and findings of these studies contribute towards more interpretable, robust, and controllable models—fundamental pillars for fostering a more human-centric development of NLP systems.

ZUSAMMENFASSUNG

Größere und komplexere Modelle haben die Messlatte in den meisten Anwendungen der *Natural Language Processing* (NLP) immer höher gelegt und sind in gesellschaftlichen Entscheidungsprozessen zunehmend präsent. Ihr Blackbox-Charakter wirft jedoch erhebliche Bedenken hinsichtlich ihrer Vertrauenswürdigkeit auf, da ihre Größe und Komplexität uns daran hindern, Modelle zu verstehen und zu kontrollieren. Als Antwort auf dieses Problem hat sich die Entwicklung von Human-Centric NLP-Modellen als Lösunganȿatz herauskristallisiert. Hierbei definieren europäische Verordnungen wichtige Anforderungen, um sicherzustellen, dass die eingesetzten Systeme mit menschlichen Werten übereinstimmen und letztendlich der Gesellschaft zugute kommen. In dieser Dissertation werden acht Studien vorgestellt, die die Verwendung von Modellerklärungen untersuchen, um die drei Schlüsselanforderungen der (1) *Interpretierbarkeit*, (2) *Robustheit* und (3) *menschliche Überwachung* von Modellen zu erfüllen. Konkret leisten wir einen Beitrag, indem wir bestehende Explainability-Methoden überprüfen, ihre Anwendbarkeit auf NLP bewerten und Ansätze entwickeln, die auf (multimodale) NLP-Eingaben in kontextspezifischen Anwendungen zugeschnitten sind. Wir zeigen, dass Modellerklärungen starke Signale enthalten, die eine explizite und modellagnostische Erkennung der Manipulation auf Text-Eingaben ermöglichen. Schlussendlich stellen wir eine Plattform für die Interaktion zwischen Mensch und Modell vor, die es Annotatoren ermöglicht, die eingesetzten Modelle zu beeinflussen und zu kontrollieren, indem sie Modellerklärungen bearbeiten und so menschliches Feedback geben können. Die Erkenntnisse und Ergebnisse dieser Studien tragen zu besser interpretier- und kontrollierbaren, sowie robusteren Modellen bei. Diese Eigenschaften sind Grundpfeiler für eine stärker auf den Menschen ausgerichtete Entwicklung von NLP-Systemen.

PUBLICATIONS

This dissertation is based on the following publications that are relevant for examination[1] and are marked with • in the dissertation.

Mosca, Edoardo, Shreyash Agarwal, Javier Rando Ramırez, and Georg Groh (May 2022). ““That Is a Suspicious Reaction!”: Interpreting Logits Variation to Detect NLP Adversarial Attacks.” In: *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Dublin, Ireland: Association for Computational Linguistics, pp. 7806–7816. doi: 10.18653/v1/2022.acl-long.538. url: https://aclanthology.org/2022.acl-long.538.

Mosca, Edoardo, Katharina Hermann, Tobias Eder, and Georg Groh (July 2022). “Explaining Neural NLP Models for the Joint Analysis of Open-and-Closed-Ended Survey Answers.” In: *Proceedings of the 2nd Workshop on Trustworthy Natural Language Processing (TrustNLP 2022)*. Seattle, U.S.A.: Association for Computational Linguistics, pp. 49–63. doi: 10.18653/v1/2022.trustnlp-1.5. url: https://aclanthology.org/2022.trustnlp-1.5.

Mosca, Edoardo, Ferenc Szigeti, Stella Tragianni, Daniel Gallagher, and Georg Groh (Oct. 2022). “SHAP-Based Explanation Methods: A Review for NLP Interpretability.” In: *Proceedings of the 29th International Conference on Computational Linguistics*. Gyeongju, Republic of Korea: International Committee on Computational Linguistics, pp. 4593–4603. url: https://aclanthology.org/2022.coling-1.406.

Mosca, Edoardo, Maximilian Wich, and Georg Groh (June 2021). “Understanding and Interpreting the Impact of User Context in Hate Speech Detection.” In: *Proceedings of the Ninth International Workshop on Natural Language Processing for Social Media*. Online: Association for Computational Linguistics, pp. 91–102. doi: 10.18653/v1/2021.socialnlp-1.8. url: https://aclanthology.org/2021.socialnlp-1.8.

[1] in accordance with Exhibit 6 of the regulations for the award of doctoral degree

The following publications contributed to the dissertation. They are not formally relevant for examination[2] and are marked with † in the dissertation.

Huber, Lukas, Marc Alexander Kühn, Edoardo Mosca, and Georg Groh (May 2022). "Detecting Word-Level Adversarial Text Attacks via SHapley Additive exPlanations." In: *Proceedings of the 7th Workshop on Representation Learning for NLP*. Dublin, Ireland: Association for Computational Linguistics, pp. 156–166. DOI: 10.18653/v1/2022.repl4nlp-1.16. URL: https://aclanthology.org/2022.repl4nlp-1.16.

Mosca, Edoardo, Daryna Dementieva, Tohid Ebrahim Ajdari, Maximilian Kummeth, Kirill Gringauz, and Georg Groh (2023). "IFAN: An Explainability-Focused Interaction Framework for Humans and NLP Models." In: *arXiv preprint arXiv:2303.03124*. (Accepted at AACL, Nov 2023). URL: https://arxiv.org/abs/2303.03124.

Mosca, Edoardo, Defne Demirtürk, Luca Mülln, Fabio Raffagnato, and Georg Groh (May 2022). "GrammarSHAP: An Efficient Model-Agnostic and Structure-Aware NLP Explainer." In: *Proceedings of the First Workshop on Learning with Natural Language Supervision*. Dublin, Ireland: Association for Computational Linguistics, pp. 10–16. DOI: 10.18653/v1/2022.lnls-1.2. URL: https://aclanthology.org/2022.lnls-1.2.

Wich, Maximilian, Edoardo Mosca, Adrian Gorniak, Johannes Hingerl, and Georg Groh (2021). "Explainable abusive language classification leveraging user and network data." In: *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*. Springer, pp. 481–496. URL: https://2021.ecmlpkdd.org/wp-content/uploads/2021/07/sub_663.pdf.

[2] in accordance with Exhibit 6 of the regulations for the award of doctoral degree

ACKNOWLEDGMENTS

Bachelor's, Master's, and now Doctorate, here we are once again at the most challenging page to write. It has been a wonderful journey, of highs and lows, with relationships and friendships that ended while new ones were born. I've had the privilege of crossing paths with many extraordinary individuals, and yes, some not so extraordinary. I've strived to give my best throughout, and I hope that most people are glad to have met me.

To my amazing supervisor, my supportive colleagues, and my invaluable mentors and co-authors, thanks for your guidance, your patience, and for having my back when I most needed it. Beyond my work, you also shaped the person I am today. To my family, thank you for your unending support and love, your sacrifices have not gone unnoticed, and I am eternally grateful. To my friends, both old and new, thank you for being my sanctuary outside of work and being there to pick up my pieces when things did not go according to plan.

This journey would not have been the same without each and every one of you. I feel extremely lucky to have you on my team. Thank you.

CONTENTS

1 Introduction 1

1.1 Overview and Motivation . . . 1

1.2 Problem Statement . . . 2

1.3 Contribution and Research Objectives . . . 3

1.4 Structure . . . 5

2 Background (XAI) 7

2.1 On its Crucial Role in Human Society . . . 7

2.2 Terminology and Taxonomies . . . 9

2.3 Interpretability in Natural Language Processing . . . 12

2.4 Evaluating Explanations . . . 16

2.5 XAI for Human-in-the-Loop . . . 19

2.6 XAI for Model Robustness . . . 23

3 Presented Studies 25

3.1 Assessing the Applicability of Explainability Approaches to NLP . . . 26

3.1.1 Motivation . . . 26

3.1.2 Study I • . . . 27

3.1.3 Discussion . . . 29

3.2 Tailoring Explainers to NLP Inputs . . . 30

3.2.1 Motivation . . . 30

3.2.2 Study II † . . . 31

3.2.3 Discussion . . . 33

3.3 Extending Explanations to Context-Aware Applications . . . 34

3.3.1 Motivation . . . 34

3.3.2 Study III • . . . 35

3.3.3 Study IV † . . . 37

3.3.4 Study V • 39
3.3.5 Discussion 42
3.4 Detecting Adversarial Attacks via Model Explanations 43
3.4.1 Motivation 43
3.4.2 Study VI † 44
3.4.3 Study VII • 46
3.4.4 Discussion 48
3.5 Enabling Model Controllability through Human Explanations 49
3.5.1 Motivation 49
3.5.2 Study VIII † 50
3.5.3 Discussion 54
4 Discussion 55
4.1 Interpretability 55
4.2 Robustness 57
4.3 Human Oversight 58
5 Conclusion 61

Appendix

A Publications • 65
A.1 Study I 65
A.2 Study III 78
A.3 Study V 92
A.4 Study VII 109
B Publications † 123
B.1 Study II 123
B.2 Study IV 132
B.3 Study VI 150
B.4 Study VIII 163
Bibliography 181

LIST OF FIGURES

Figure 1.1 Breakdown of this work into its three main goals and their corresponding objectives. 4

Figure 2.1 Information flow in transparent models versus post-hoc explainability settings. 10

Figure 2.2 Visual taxonomy for post-hoc explainability methods. 11

Figure 2.3 A practical overview of the settings for evaluating explanations, compared across different metrics. 19

Figure 2.4 General EBDH pipeline for NLP models, including the potentially faulty model, the human annotators offering feedback, and the three-step workflow. 22

Figure 3.1 An overview of the conducted studies, together with how they address this work's research objectives. 25

Figure 3.2 Instructive explanation examples provided in study I for sentiment analysis. 28

Figure 3.3 Overview of GrammarSHAP's pipeline. 32

Figure 3.4 Explanation comparison at different granularity levels and between three explainers for grouped features relevance. . . . 33

Figure 3.5 Hate speech detection architectures for the text model and the social model. 36

Figure 3.6 Explanations produced by the text, history, and network submodels. 39

Figure 3.7 Multi-modal model classification architecture—combining both text and categorical features. 41

Figure 3.8 Overview of the pipeline once the adversarial detector is ready to be deployed together with the steps required to train it. . . 45

Figure 3.9 Overall structure of the IFAN's platform. 51
Figure 3.10 Access levels to IFAN's functionalities and architecture of the integrated NLP models. 52
Figure 3.11 Comparison between fine-tuning a model with and without feedback rebalancing. 53

ACRONYMS

AI	Artificial Intelligence
EBHD	Explanation-Based Human Debugging
HitL	Human in the loop
IFAN	Interaction framework for artificial and natural intelligence
LLMs	Large language models
ML	Machine learning
NLP	Natural language processing
SHAP	Shapley additive explanations
WDR	Word-level differential reaction
XAI	Explainable artificial intelligence

1

INTRODUCTION

1.1 OVERVIEW AND MOTIVATION

Artificial Intelligence (AI) systems are at the core of the current technological and societal revolution (European Commission, 2020; West, 2018). In particular, technologies based on *Natural Language Processing* (NLP) have witnessed a tremendous growth thanks to an increase in data availability, computational resources, research efforts, and funding (Ignat et al., 2023). Transformers, diffusion architectures, and more in general *Large Language Models* (LLMs) are the undisputed protagonists of the latest developments in the field (Brown et al., 2020; Scao et al., 2022). They are setting new standards in performance and versatility, solving tasks previously deemed challenging or even unapproachable (B. Min et al., 2021; W. X. Zhao et al., 2023). At the same time, they can produce natural language artifacts that rival—and occasionally even surpass—human-level quality (OpenAI, 2022).

Employing large models, however, has also considerable drawbacks concerning their interpretability and trustworthiness (Arrieta et al., 2020; Zachary C Lipton, 2016; W. James Murdoch et al., 2019). Indeed, the high number of parameters and architectural complexity make them behave like black-boxes, hindering our ability to understand and control them. This lack of transparency stands as a major obstacle regarding their adoption and integration into human and societal processes (Molnar, 2019).

Working towards *transparent* and *trustworthy* NLP has thus become a priority, with even European regulations setting *"a human-centric approach to AI"* (European

Commission, 2019, section 2) as the primary goal to ensure that intelligent algorithms act in line with human values and ultimately benefit society at large.

It is crucial to act timely, as current research progress plays a pivotal role in shaping how AI and human society will coexist in the future. In this context, research and insights from *eXplainable Artificial Intelligence* (XAI) (Arrieta et al., 2020; Molnar, 2019) warrant attention as they offer the potential to overcome model opacity—enabling us to trust, interpret, and productively control NLP models.

1.2 PROBLEM STATEMENT

In the context of NLP models, human-centricity is not a monolithic concept. It is instead a complex notion, entailing a wide range of factors and criteria that need to be fulfilled. Current European regulations also steer away from a single precise definition, opting instead to provide a list of (seven) key requirements for the future design, development, and deployment of AI systems (European Commission, 2019, 2020). This dissertation focuses on three of these aspects: i.e. (1) *interpretability*, (2) *robustness*, and (3) *human oversight*.

Interpretability refers to the ability to understand and explain the decisions made by an NLP model (more details in 2) (Arrieta et al., 2020; Molnar, 2019). Despite the impressive capabilities of state-of-the-art architectures, their decision-making process is opaque, hard to comprehend, and often diverging from human reasoning (Guidotti et al., 2019; Madsen, Reddy, and Chandar, 2022). Lacking interpretability can lead to mistrust and hinder the adoption of such systems in various sectors. This is especially true in high-stakes domains—e.g. medicine (Locke et al., 2021; Yuqing Wang, Y. Zhao, and Petzold, 2023)—where understanding the reasoning behind decisions is crucial (Molnar, 2019). Furthermore, the inability to interpret these models may pose ethical and legal challenges as it becomes difficult to hold anyone accountable for wrong decisions and model misbehavior (European Parliament, 2016; Wachter, Mittelstadt, and Floridi, 2017).

The second aspect, **robustness**, is about ensuring that NLP models perform reliably under different conditions and are resilient to attacks and manipulated inputs (W. E. Zhang et al., 2020). Although recent advancements have made models less reliant on small domain-specific corpora and specific input patterns, they are still sensitive to slight input changes (Xuezhi Wang, H. Wang, and Yang, 2022). This sensitivity can be exploited by adversarial third parties, who can craft inputs with the intention of fooling deployed models, leading to unpredictable and erroneous outputs (Garg and Ramakrishnan, 2020; Ren et al., 2019). Lacking robustness is thus not a viable option when safety is paramount, making developing defense mechanisms essential for deployment (European Commission, 2020; Yuan et al., 2019).

Finally, **human oversight** refers to the necessity for human involvement and control in the decision-making process of NLP models (Monarch, 2021; Z. J. Wang et al., 2021). While intelligent systems have the potential to automate and streamline many tasks, it is vital that humans remain in the loop to ensure that the outcome aligns with our values and societal norms (European Commission, 2020). Human oversight can be seen as an extension of interpretability. Beyond making the models more transparent and understandable, it also involves designing interfaces and workflows that allow humans to interact with the models in a meaningful and productive way (Lertvittayakumjorn and Toni, 2021). Moreover, it places humans back in a position of control, therefore restoring human agency in NLP processes and enhancing accountability.

1.3 CONTRIBUTION AND RESEARCH OBJECTIVES

This dissertation investigates methods and strategies to improve (1) *interpretability*, (2) *robustness*, and (3) *human oversight*—ultimately working towards a more human-centric development of NLP models. Our methodology is deeply rooted in the field of XAI and employs model explanations to pursue all three goals. As such main goals are broad and complex, we break them down into five more specific objectives. The first three work towards enhancing interpretability, whereas the fourth and fifth contribute to improving robustness and human oversight respectively.

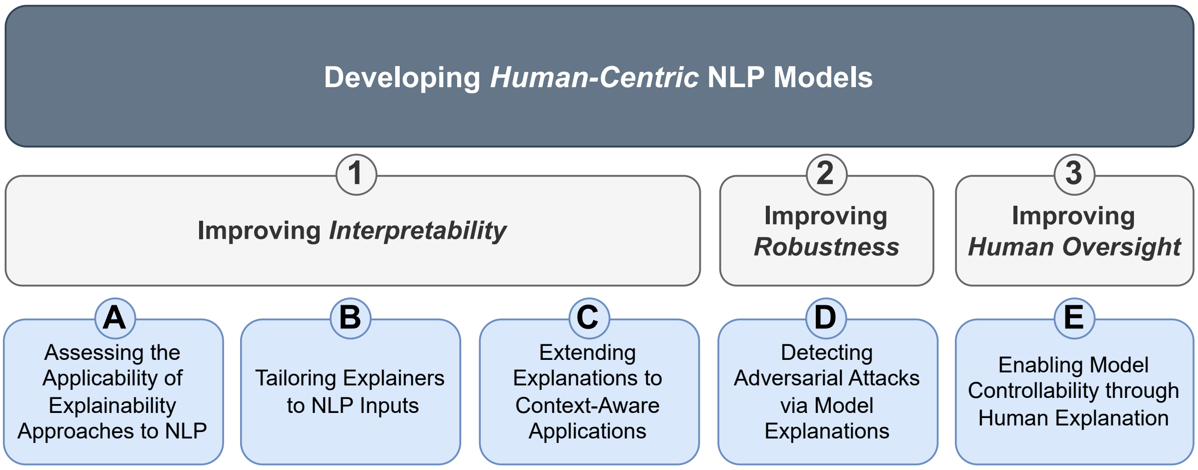

Developing *Human-Centric* NLP Models
1
Improving *Interpretability*
2
Improving *Robustness*
3
Improving *Human Oversight*
A
Assessing the Applicability of Explainability Approaches to NLP
B
Tailoring Explainers to NLP Inputs
C
Extending Explanations to Context-Aware Applications
D
Detecting Adversarial Attacks via Model Explanations
E
Enabling Model Controllability through Human Explanation

Figure 1.1: Breakdown of this work into its three main goals and their corresponding objectives.

(A) - ASSESSING THE APPLICABILITY OF EXPLAINABILITY APPROACHES TO NLP: As XAI extends beyond the NLP field, this objective aims at investigating existing XAI methods to understand their underlying principles, limitations, and effectiveness for NLP models and text data. This can shed light on the landscape of available approaches and their potential added value for NLP while simultaneously aiding in the discovery of lesser-known, more suitable interpretability techniques.

(B) - TAILORING EXPLAINERS TO NLP INPUTS: This objective focuses on customizing explainability approaches to better suit the unique characteristics of NLP inputs. It involves developing or modifying existing explainability techniques to account for the particularities of language data—such as sequential structure, context-dependency, and semantic complexity.

(C) - EXTENDING EXPLANATIONS TO CONTEXT-AWARE APPLICATIONS: In many real-world NLP use cases context is paramount and the best-performing models rely on multiple input modes rather than just text. This objective addresses the challenge of providing meaningful explanations for (often multi-modal) NLP models operating in context-aware applications. Furthermore, it aims at providing deeper insights into how different data sources interact and how context shapes the deployed models' behavior.

(D) - DETECTING ADVERSARIAL ATTACKS VIA MODEL EXPLANATIONS: This objective involves leveraging model explanations to mitigate the issue of adversarial text attacks. Beyond making the model more robust, the aim is to develop methods that can explicitly identify attacking attempts and thus build an additional layer of defense against adversarial agents.

(E) - ENABLING MODEL CONTROLLABILITY THROUGH HUMAN EXPLANATIONS: The aim is to build pipelines and approaches that utilize explainability to facilitate human oversight and control over NLP models. More specifically, (non-technical) stakeholders should be able to understand the "*why*" behind the model's predictions and effectively provide human feedback to steer its behavior.

We take these five objectives as the foundation for the eight studies conducted in the context of this dissertation, collectively contributing to more interpretable, robust, and controllable NLP models.

1.4 STRUCTURE

We structure the rest of this dissertation as follows. Chapter 2 revolves around the background of this work, i.e. a broad and comprehensive overview of the field of XAI. Chapter 3, instead, organizes and presents the eight studies that contribute to this work together with their motivation, contribution, methodology, and limitations. For clarity, we mark with • studies that are relevant for the examination. The remaining ones, marked with †, are not formally relevant to the examination but still contributed to this thesis. Chapter 4 provides an overarching discussion—highlighting our successes, limitations, and key takeaways for future work. Finally, Chapter 5 summarizes and concludes this work.

2

BACKGROUND (XAI)

eXplainable Artificial Intelligence (XAI) is an emerging research field concerned with understanding the *"why"* behind the behavior of otherwise opaque AI systems. In practice, XAI primarily focuses on interpreting decisions and predictions from *Machine Learning* (ML) algorithms—in particular deep learning models.

In the following sections, we discuss the current state of XAI regarding its crucial role in AI's societal adoption (section 2.1), the currently used terminology and existing taxonomies (section 2.2), the main explanation types for NLP models (section 2.3), as well as approaches and challenges w.r.t. evaluating explanations (section 2.4). Furthermore, we take a brief look at the current literature on XAI for human-in-the-loop systems (section 2.5) and model robustness (section 2.6).

As we progress through the chapter, we will reference instructive examples of XAI approaches. We do so without delving into details as they are not central to our methodology and do not contribute to this chapter's intended scope. We do however highlight the significance of the SHAP framework (S. M. Lundberg and Lee, 2017) in the context of our research, which is comprehensively described in our first study (Study I, Appendix A.1). The sections 2.1, 2.2, and 2.4 draw inspiration from Mosca (2020).

2.1 ON ITS CRUCIAL ROLE IN HUMAN SOCIETY

The growing presence of AI systems in human society, particularly in decision-making processes, is closely tied to research efforts in XAI (Arrieta et al., 2020). As AI solutions are increasingly being adopted in sensitive sectors—e.g. legal (Marques et al., 2019), medical (A. S. Lundervold and A. Lundervold, 2019), and mobility (Z. Zhang, 2021)—the literature identifies several contexts that render XAI essential:

SOCIETAL ACCEPTANCE: AI systems that do not offer a human-interpretable rationale for their conclusions are unlikely to be accepted by society (Goodman and Flaxman, 2017; Molnar, 2019). Neglecting the *"why"* behind a decision strongly challenges key components of human nature—curiosity and desire to learn and understand (Molnar, 2019). Thus, XAI fosters societal acceptance and trust in AI as explanations help us make sense of how an artificial agent produced a specific output (Kim, Rudin, and J. A. Shah, 2014; Rudin and Ustun, 2018).

ALIGNMENT WITH SCIENTIFIC OBJECTIVES: The black-box nature of many ML algorithms can also impede discovering the underlying mechanisms behind observed phenomena (Arrieta et al., 2020). We feed large amounts of data and we observe lots of outputs, yet we do not comprehend the transformation process in between. XAI has the potential to transform models into a source of knowledge and thus align the field with the main goal of science: understanding (Molnar, 2019).

COMPLIANCE WITH LEGAL GUIDELINES: The European Union *General Data Protection Regulation* (GDPR) (European Parliament, 2016), *White Paper on Artificial Intelligence* (European Commission, 2020), and the commission's communication on *Building Trust in Human-Centric Artificial Intelligence* (European Commission, 2019) are key examples of regulatory frameworks regarding the interpretability of AI. The former, for instance, introduces the highly debated *"right to an explanation"* (Edwards and Veale, 2017; Wachter, Mittelstadt, and Floridi, 2017; Wachter, Mittelstadt, and Russell, 2017), whereas the latter strongly advocates to use interpretability in AI as a means to ensure transparency, accountability, and human oversight.

IDENTIFYING AND DEBUGGING FAULTY SYSTEMS: Machine learning models are renowned for picking up—and at times even amplifying—biases. At the same time, they can produce hallucinations and other unintended artifacts (Bolukbasi et al., 2016;

Z. Ji et al., 2022). Explanations can expose these unwanted behaviors (Mosca, Wich, and Groh, 2021), shed light on their causing factors, and in some cases help mitigate or even correct them (Lertvittayakumjorn and Toni, 2021).

2.2 TERMINOLOGY AND TAXONOMIES

Various terms such as *Explainability*, *Transparency*, *Comprehensibility*, and *Interpretability* are often used interchangeably in the literature. However, they do not carry the same meaning (Zachary C Lipton, 2016).

Interpretability is quite clearly not a monolithic concept in machine learning. It is instead an umbrella term that encompassess a variety of ideas (Zachary C Lipton, 2016). Within the literature, we often encounter broad definitions offering limited practical utility. For instance, Arrieta et al. (2020, Page 5) defines it as *"the ability to explain or to provide the meaning in understandable terms to a human"*.

However, other works such as W. James Murdoch et al. (2019) and Zachary C Lipton (2016) prefer to distinguish between two classes of interpretable systems, i.e. *transparent models* and *post-hoc explainability* (see Figure 2.1). The former describes ML models that were specifically designed to be interpretable. The latter, instead, encompasses all techniques actively explaining algorithms that are not inherently interpretable. Also the term *post-hoc*—i.e. *"after the fact"*—precisely describes the explainability process occurring after the model has already been designed and trained for its intended task.

Concerning transparent models, the literature defines several levels to which a model is inherently interpretable. These range from being entirely simulatable or replicable by a human at once (i.e. *simulability*) to at least being understandable in each of its single components (i.e. *decomposability*) (Arrieta et al., 2020; Zachary C Lipton, 2016).

Post-hoc explainability, instead, is not a model property and refers to utilizing an additional *explanation method* to explain the behavior of non-transparent models (Arrieta et al., 2020). While this setting can convey useful—and at times essential—

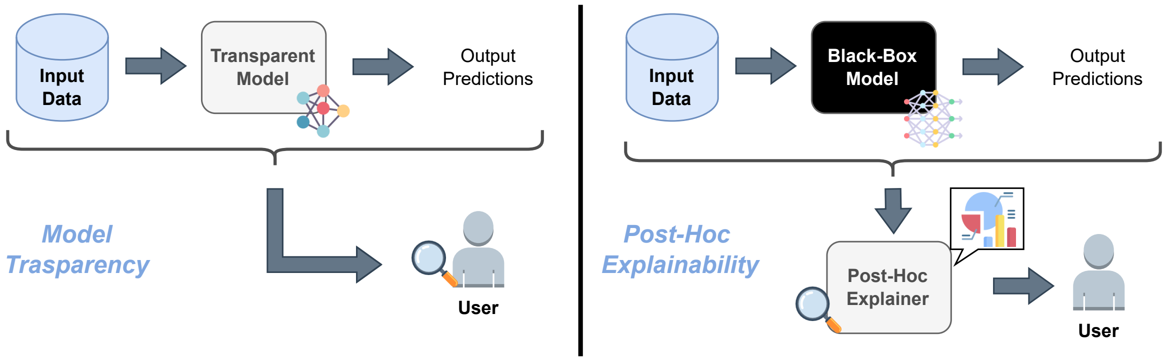


Figure 2.1: Comparative illustration of the information flow in transparent models versus post-hoc explainability settings. When the model is transparent, stakeholders can directly interpret the model and understand the outcome. When the model is black-box and thus not inherently interpretable, an external explainability technique is used subsequently to produce reasons for the model's output.

information about the model, it usually does not suffice to fully comprehend the inner workings of complex architectures. Nevertheless, opting to use post-hoc approaches has the major advantage of not implying any restriction for the model's architecture and hence can be used without any sacrifice in model performance.

The literature categorizes post-hoc approaches under a variety of aspects—e.g. which models they can be applied to, the input and model components that the produced explanations refer to, and the format in which explanations are presented to humans (Arrieta et al., 2020; Doshi-Velez and Kim, 2017; Guidotti et al., 2019; Madsen, Reddy, and Chandar, 2022; Mittelstadt, Russell, and Wachter, 2019; W. James Murdoch et al., 2019). A visual summary for such categorization is illustrated in Figure 2.2.

MODEL SPECIFIC OR MODEL AGNOSTIC: Model-specific explainability techniques are only applicable to specific architectures or a specific class of models. They often offer higher accuracy and computational efficiency by taking advantage of assumptions specific to the model type they are tailored to. Two examples are TreeSHAP (S. M. Lundberg, G. Erion, et al., 2020) and DeepSHAP (S. M. Lundberg and Lee, 2017), which are built exclusively for decision trees and deep neural networks respectively. Model-agnostic methods, instead, do not prescribe any requirements for the model they explain and can therefore provide explanations about any architecture without making any assumption on its characteristics. LIME (M. T. Ribeiro, S. Singh, and Guestrin, 2016) is a framework with a strong focus on model agnosticism that has gained great popularity thanks to its "*plug and play*" design.

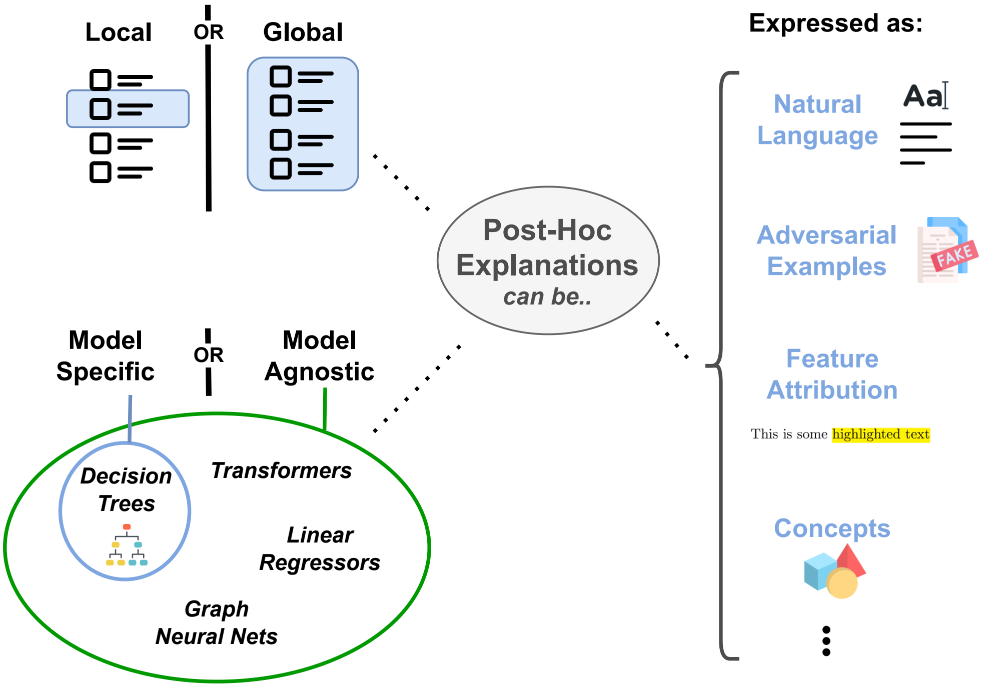


Figure 2.2: Visual sketch of the taxonomy for post-hoc explainability methods. The various parts of the figure represent the key aspects of post-hoc explainability methods, including their *scope* (global or local), *model applicability* (agnostic or specific), and a few examples for the diverse range of options in terms of *explanation format*.

LOCAL OR GLOBAL: These terms refer to prediction-level and dataset-level explanations respectively (Zachary C Lipton, 2016). Local approaches provide reasons regarding one specific model prediction. For instance, LIME (M. T. Ribeiro, S. Singh, and Guestrin, 2016) and DeepLIFT (Shrikumar, Greenside, and Kundaje, 2017) quantify the relevance of each feature w.r.t. a single input instance. On the other hand, global methods aim at producing information regarding the overall model's behavior in terms of what patterns and rules it has learned. SAGE (Covert, S. M. Lundberg, and Lee, 2020), for instance, can quantify the predictive power of each feature in the dataset for a given model.

EXPLANATION FORMAT: Interpretability techniques vary significantly in terms of how the produced explanations are presented (Doshi-Velez and Kim, 2017). Among the most common types we can find *input feature attribution*, *influential samples*, *counterfactuals*, and *natural language rationales* (Bhatt et al., 2020; Madsen, Reddy, and Chandar, 2022). See the following section (2.3) for further details about common formats in NLP.

As for this work's scope, we observe that the most widely used models are inherently black-box (Ignat et al., 2023), making it unlikely for stakeholders to accept sacrificing performance when introducing interpretability features. Therefore, our primary focus is centered on post-hoc explainability techniques as they provide a larger utility within this context. While our studies predominantly involve local explanations, it's worth noting that, in certain instances, we have also incorporated global approaches as a complement.

As for the terminology used, we refrain from using the term *model transparency* as its definition does not describe our methodology. Instead, we adopt *post-hoc explainability*, at times either shortened to *explainability* or encompassed by the broader term of *interpretability* (Arrieta et al., 2020).

2.3 INTERPRETABILITY IN NATURAL LANGUAGE PROCESSING

Although XAI initially gained its popularity in computer vision, the advent of complex neural NLP models (Brown et al., 2020; Devlin et al., 2019; Y. Liu et al., 2019) has driven the demand for interpretability methods also for text-focused applications such as dialog systems, text classification, and summarization.

Several surveys (Belinkov and Glass, 2019; Danilevsky et al., 2020; Sun et al., 2021)—as well as scientific tutorials at leading conferences (Belinkov, Gehrmann, and Pavlick, 2020; E. Wallace, Gardner, and S. Singh, 2020)—offer a broad overview of the application of XAI methods to NLP. Following a similar motivation to this work, Madsen, Reddy, and Chandar (2022) focuses on post-hoc explainability approaches in NLP. As one can observe, in practice, certain explanation types have attracted substantially more attention than others (Bhatt et al., 2020).

FEATURE ATTRIBUTION: They are also known as *saliency maps* and *feature relevance* scores. These explanations address the question "*which input tokens are most important for the prediction?*" (Madsen, Reddy, and Chandar, 2022) and are by far the most commonly employed techniques (Bhatt et al., 2020). Notorious examples are LIME (M. T. Ribeiro, S. Singh, and Guestrin, 2016) and SHAP (S. M. Lundberg and Lee, 2017). More NLP-focused variants also exist, such as HEDGE (H. Chen, Zheng, and Y. Ji, 2020) and LS-Tree (J. Chen and Jordan, 2020).

Feature attribution approaches are highly adaptable as the input features are always available and generally meaningful to humans. However, they are usually limited to producing a score for each feature and for a single class (Madsen, Reddy, and Chandar, 2022). This often implies the necessity of repeating the procedure at each time step in sequence-to-sequence applications (Jiwei Li et al., 2016).

INFLUENTIAL SAMPLES: Such approaches select samples from the dataset that—at least from the model's perspective—are closely related to the current input and thus should lead to a similar output (Madsen, Reddy, and Chandar, 2022). Methods such as *influential functions* (Koh and Liang, 2017) and *TracIn* (G. Pruthi et al., 2020) are part of this category and indeed answer "*which training examples are most influential for the outcome?*"

Influential samples explanations are also particularly useful for uncovering dataset-level artifacts such as mislabeled samples or incorrectly pre-processed texts.

ADVERSARIAL EXAMPLES: Generating adversarial examples addresses the question "*what input would fool the model into producing an incorrect prediction?*" and can expose model weaknesses—revealing that even highly-performing architectures suffer from a lack of robustness. Popular adversarial attack methods to generate such examples are *HotFlip* (Ebrahimi et al., 2018) and PWWS (Ren et al., 2019). Further details on robustness and adversarial attacks for NLP can be found in section 2.6.

COUNTERFACTUALS: *"What (small) change in the input would cause a change of prediction to a predefined output?"* (Molnar, 2019) is the core of counterfactual explanations. They describe a hypothetical variation of the current input that would cause a change in prediction from the model. *Polyjuice* (Wu et al., 2021) and *MiCe* (Ross, Marasović, and Peters, 2021) are popular frameworks applicable to NLP use cases.

Counterfactual explanations—also sometimes referred to with the term *contrastive*—are highly praised by works coming from the social sciences given their similarity with the human's usage of causal implications (Byrne, 2019; P. Lipton, 1990; Miller, 2019). They are also extremely relevant for applications where model predictions should allow some form of recourse for the affected parties. For instance, if a job application is rejected due as a result of an automatic check, counterfactual explanations can point out what could be changed in order to revert the outcome.

NATURAL LANGUAGE RATIONALES: Explanations are directly generated as natural text. They do not answer any specific question, although Madsen, Reddy, and Chandar (2022) claim they address *"what would a generated natural language explanation be?"*. We argue that such question is rather general and does not prescribe any useful content guidelines. CAGE (Rajani et al., 2019) constructs rationales for a model by fine-tuning an additional GPT-2 instance (Radford et al., 2018)) on a dataset of human explanations.

Explanations in the form of natural language are very accessible and easier to understand for users not coming from the ML field. Indeed, the given explanation can be interpreted without needing to analyze numerical scores or other abstract elements—e.g.feature attributions or linguistic concepts.

Natural language explanations can also be used in a non-post-hoc system without sacrificing model performance. Several works enable the use of such rationales to intrinsically force the model to explain itself and generalize better (Camburu et al., 2018; S. Kumar and Talukdar, 2020; Hui Liu, Yin, and W. Y. Wang, 2019). However, we refrain from considering them in the *model transparency* category as the model's inner workings are still opaque.

CONCEPTS: *Concepts* are an abstraction of the input that groups and describes a class or a cluster of samples sharing common characteristics. For instance, black and white stripes can be a concept for zebras in images, while the adjectives *"happy"* and *"glad"* can function as concepts for the positive class in a sentiment analysis task. In other words, they provide an answer to *"what concepts can best represent a class?"* (Madsen, Reddy, and Chandar, 2022) by generating a list of traits and features that act as a summary for similar input instances. The overall set of concepts conveys an approximate description of the strongest themes and patterns identified by the model in the training dataset. Popular techniques are TCAV (Kim, Wattenberg, et al., 2018), NIE (Vig et al., 2020), and ACE (Ghorbani, Wexler, et al., 2019).

Some concept-explanation methods e.g. TCAV require a pre-compiled list of concepts and they quantitatively test for their representational power within the dataset. However, methods like ConceptSHAP (Yeh et al., 2020) are able to operate in an unsupervised manner and can directly produce concepts without any human guidance. Yeh et al. (2020) also defines *completeness*, a quantity measuring how well a given set of concepts is in explaining the model's behavior w.r.t. the entire dataset.

Works like Madsen, Reddy, and Chandar (2022) view concept explanations neither as local nor global, but rather belonging to their own category. While not entirely disagreeing with this view, we argue that such approaches possess a strong global character as they mostly summarize conceptual features learned at the dataset level and do not really focus on specific instances.

ENSEMBLE: Ensemble explanations are a broad category of methods that construct global explanations by combining local ones. In other words, the model behavior at the dataset level is expressed by grouping explanations referring to single prediction utterances. An instructive example is SP-LIME (M. T. Ribeiro, S. Singh, and Guestrin, 2016), which computes the overall importance of each feature by summing its relevance for every instance in the dataset or subset thereof.

Data Shapley (Ghorbani and J. Zou, 2019) and SAGE (Covert, S. M. Lundberg, and Lee, 2020) are for instance global variants for influential samples and feature attribution respectively. Indeed, they respectively identify the most valuable training samples and input features contributing to the model's predictive performance.

ATTENTION BASED: Attention is a key component of modern neural NLP architectures, most notoriously in transformers (Vaswani et al., 2017)—offering substantial improvements in performance and interpretability of s.o.t.a architectures. Despite the large debate on its validity as an explanation (Bastings and Filippova, 2020; Bibal et al., 2022; Jain and Byron C Wallace, 2019; Serrano and N. A. Smith, 2019; Wiegreffe and Pinter, 2019), some attention-based approaches have improved on using raw attention scores and are used for interpretability purposes (Abnar and Zuidema, 2020).

2.4 EVALUATING EXPLANATIONS

Measuring the validity and quality of an explanation is arguably one of the most arduous challenges currently faced by XAI research. Interpretability is defined as providing information in understandable terms to humans (Arrieta et al., 2020). Hence—unlike classical ML benchmarks (Japkowicz and M. Shah, 2011)—explanations and their evaluations can often only be expressed in a qualitative form (Doshi-Velez and Kim, 2017; Madsen, Reddy, and Chandar, 2022). Quantitative evaluations are viable only in specific cases (Bastings, Ebert, et al., 2021; Hao, 2020; Poerner, Schütze, and Roth, 2018), and there is no general agreement on how to measure interpretability in the broader case.

What makes distinguishing poor explanations from high-quality ones so challenging is the co-existence of two different criteria against which explanations are evaluated:

plausibility and *faithfulness*. Plausibility, as the name suggests, refers to whether it looks convincing to humans. Faithfulness, instead, refers to how accurately it reflects the model's true reasoning (Jacovi and Goldberg, 2020).

It's quite straightforward to notice that plausibility and faithfulness can be highly uncorrelated. Indeed, plausibility does not provide any guarantee that what the explanation conveys faithfully describes the logic of the ML model. The same holds vice-versa; even the most faithful explanation could look—and often does—implausible to humans.

When it comes to plausibility, we do not actually judge whether we believe the explanation is plausible in terms of how the model operates, but rather its similarity to the explanations we would produce as humans (Herman, 2017; Jacovi and Goldberg, 2020). Hence, we evaluate mostly based on personal beliefs and are thus affected by our cognitive bias (Miller, 2019; Nickerson, 1998).

Doshi-Velez and Kim (2017) build a standard for methods measuring interpretability—also adopted by later works (Madsen, Reddy, and Chandar, 2022; Poerner, Schütze, and Roth, 2018). The authors claim that evaluation approaches generally fall into three settings: *application grounded*, *human grounded*, and *functionally grounded*.

APPLICATION GROUNDED (REAL HUMANS, REAL TASKS): This category refers to the direct application and evaluation of an explainability system within its intended use case (Doshi-Velez and Kim, 2017). This often implies the need to involve domain experts, e.g. lawyers or other professionals in law in the case of an explainable legal NLP framework (Shukla et al., 2022).

HUMAN GROUNDED (REAL HUMANS, SIMPLIFIED TASKS): The evaluation takes place in a simplified setting compared to the target application (Doshi-Velez and Kim, 2017). Naturally, to ensure that the conducted experiments are a good evaluation proxy, the character and goals of the simpler setting should remain aligned with the original use case. For instance, Mohseni and Ragan (2018) employ non-experts to rank explanations produced for simple text classifiers.

While they are rarely able to replicate the specificity of application-grounded settings, human-grounded evaluations allow us to relax strong constraints—such as the availability of domain experts, high costs, and setup time—often required by testing directly on the target use case. Moreover, human-grounded experiments can at times provide great flexibility in terms of design factors such as task complexity (Doshi-Velez and Kim, 2017; Kim, Chacha, and J. A. Shah, 2015).

FUNCTIONALLY GROUNDED (NO HUMANS, PROXY TASKS): These approaches rely on some formal definition of model interpretability and then design a proxy task to evaluate explanation quality without involving humans (Doshi-Velez and Kim, 2017). For instance, Poerner, Schütze, and Roth (2018) evaluate explanations for text classifiers using hybrid documents. Briefly, they concatenate inputs belonging to different classes and then measure how often feature attribution methods deem relevant the part of the input corresponding to the predicted class.

Functionally-grounded approaches are the ideal scenario in terms of scalability and cost-effectiveness as they bypass the need for humans. However, good definitions for explanation quality and proxy tasks that reflect those concepts are hard to construct formally (Doshi-Velez and Kim, 2017). Hence functionally-grounded experiments are particularly challenging to design and are usually applicable only to a very restricted set of use cases.

Figure 2.3 visually summarizes and compares the described settings across different dimensions. Doshi-Velez and Kim (2017) claim that application-grounded settings are well-aligned with the goals of interpretability, i.e. ensuring that the ML system delivers on its intended purpose. However, we argue that this is only guaranteed for explanation plausibility and utility as long as the assessment is made by humans, even if they are domain experts (Herman, 2017; Jacovi and Goldberg, 2020). The same can be argued for most human-grounded assessments.

Functionally-grounded settings, on the other hand, are more prone to evaluate faithfulness as long as the proxy is well-designed. At the same time, no human involvement can result in assessments that are highly uncorrelated with plausibility

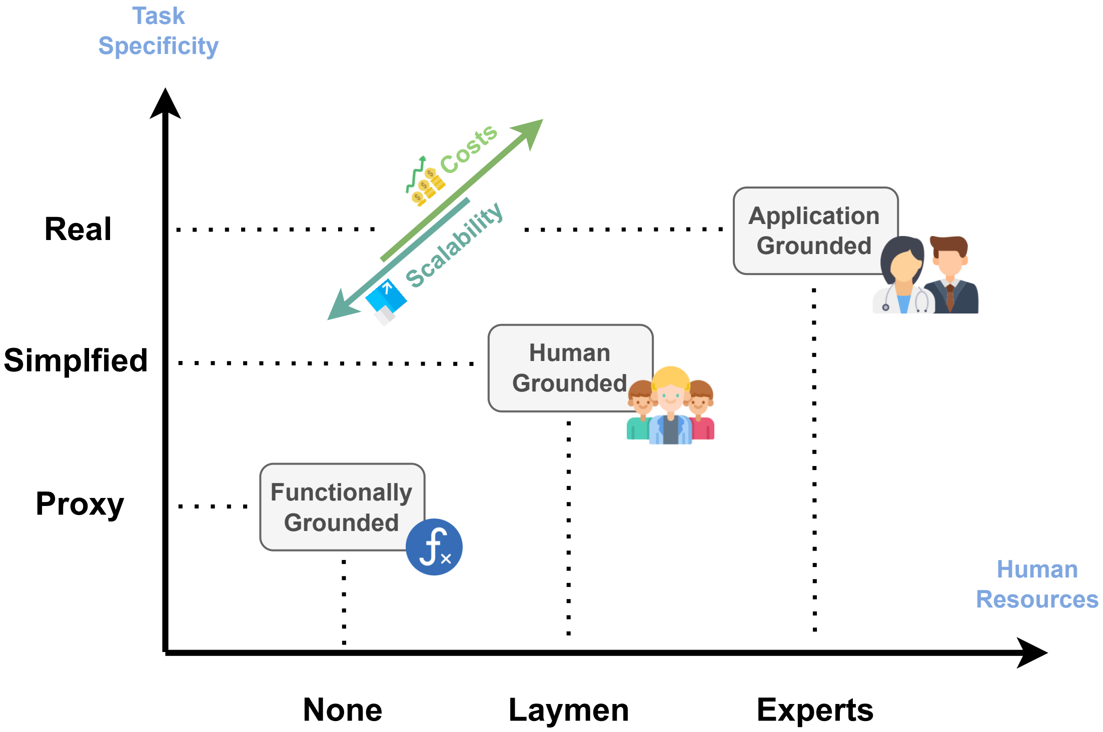


Figure 2.3: Evaluating explanations: a practical overview of the three settings described by Doshi-Velez and Kim (2017), compared across different metrics. The x-axis represents the amount of human resources required to conduct the experiments, while the y-axis indicates the level of specificity associated with the evaluation task. Naturally, both dimensions also have a direct impact on implementation costs and scalability. Figure inspired by Mosca (2020, Figure 5).

measurements. Ultimately, deciding on which setting is the most appropriate to use should strongly consider factors like costs, application constraints, need for human resources, ease of generating proxy tasks/metrics, and priority of either faithfulness or plausibility.

2.5 XAI FOR HUMAN-IN-THE-LOOP

As already discussed in 2.1, XAI research plays a key role in societal acceptance, alignment with scientific objectives, and compliance with legal guidelines and requirements. Moreover, explanations naturally foster the interaction between humans and ML systems. Indeed, they have been utilized to support decision making (Knapič et al., 2021; Lertvittayakumjorn, Petej, et al., 2021), promote trust in AI systems (Jacovi, Marasović, et al., 2021; Ras, Gerven, and Haselager, 2018), and even teach humans how to improve in performing challenging tasks (Lai, Han Liu, and Tan, 2020).

Especially when it comes to *human oversight* on AI systems, interpretability can shed light on models' unwanted behavior and whether they serve the purpose intended by the designers. Hence, explanations benefit humans w.r.t. understanding the model and making sure that its employment is beneficial rather than harmful (Ribera and Lapedriza, 2019). Beyond that, research has also explored whether interacting with humans can also be beneficial for ML algorithms.

The field of *Human-in-the-Loop* (HitL) studies methods for humans and machines to work together effectively (Monarch, 2021). More specifically, in the context of ML, it explores how continuous human-model interactions—even after deployment—can help improve systems and their predictions.

While a large chunk of the literature deals with classical *active learning* aspects such as sampling, collecting annotations, and online data augmentation, some works explore how human explanations can provide useful feedback to models (Lertvittayakumjorn and Toni, 2021). In fact, works like Ray et al. (2019), Selvaraju et al. (2019), and Strout, Y. Zhang, and Mooney (2019) show improvements in performance and interpretability when models are provided with human rationale during learning or at later stages of deployment.

Combining explanations with HitL to debug and improve models (Han, Byron C. Wallace, and Tsvetkov, 2020; Z. J. Wang et al., 2021) is referred to as *Explanation-Based Human Debugging* (EBHD) by Lertvittayakumjorn and Toni (2021). The authors also thoroughly review existing approaches within NLP and categorize them based on multiple aspects. These are visually organized in Figure 2.4 and include:

(BUG) CONTEXT: It refers to the situation that the HitL mechanism is aiming to fix or improve. This includes the inspected model, the bug source, and its intended NLP task—e.g. natural language inference (Zylberajch, Lertvittayakumjorn, and Toni, 2021) or question answering (M. T. Ribeiro, S. Singh, and Guestrin, 2018).

WORKFLOW: It describes the procedure adopted to refine the model in its context which consists of three sequential steps: (1) explanations are presented to the human annotators, (2) human feedback is collected, and (3) the model is updated based on the human rationale. The debugging workflow can be applied iteratively to further improve the model. In this case, the explanations are expected to change as the model gets updated multiple times.

EXPERIMENTAL SETTING: The mode in which humans have been involved to provide annotations and feedback. This ranges from experiments being carried out with annotators in-person (Kulesza et al., 2009) to leveraging crowdsourcing platforms (Smith-Renner et al., 2020) or even simulating human feedback (Teso and Kersting, 2019).

As a concrete example, Yao et al. (2021) uses explanations to debug a BERT (Devlin et al., 2019) and a RoBERTa (Y. Liu et al., 2019) instance. Specifically, annotators are presented with hierarchical feature attributions to which they provide refinement suggestions as natural language. The feedback is transformed into first-order logic rules used to condition learning for new (unlabeled) samples.

Lertvittayakumjorn, Specia, and Toni (2020) propose FIND, a framework using global explanations to extract and display the most relevant lexicon features used by the model. Human annotators can disable irrelevant hidden features to reduce prediction artifacts in text classifiers. FIND was shown to be particularly effective against significant bugs—such as gender bias in language detection (Lertvittayakumjorn and Toni, 2021).

SEARs from M. T. Ribeiro, S. Singh, and Guestrin (2018) is suitable instead to find more subtle model-specific bugs. It can construct universal replacement rules that result in adversarial samples—and thus wrong predictions on several instances. The utility of SEARs has been shown on multiple NLP tasks, ranging from machine comprehension to visual question-answering.

When it comes to human annotators, several factors play a role in the quality of the feedback—*model understanding*, *willingness* (to contribute), *trust*, *frustration* (especially

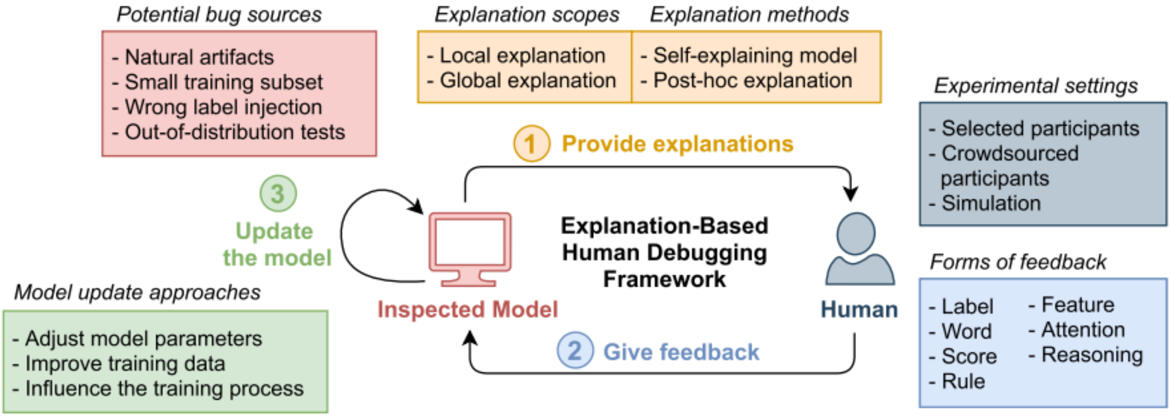


Figure 2.4: General EBDH pipeline for NLP models, including the potentially faulty model, the human annotators offering feedback, and the three-step workflow. For clarity, a (colored) box is provided for all components, enumerating examples typically encountered in practice. Figure originally from Lertvittayakumjorn and Toni (2021, Figure 1).

when interacting with poor models), and *expectations* (in seeing the model improving) (Amershi et al., 2014; Lertvittayakumjorn and Toni, 2021). Analogously to evaluation frameworks for explainability, there is a clear trade-off between feedback quality and scalability of a setting in terms of how many annotations can be collected. In-person subjects are naturally more committed and involved but only allow for small experiments (Lertvittayakumjorn and Toni, 2021). Crowdsourcing and simulation offer substantially open the door to larger experiments but they imply risks and drawbacks in terms of the meaningfulness of the responses (Lertvittayakumjorn and Toni, 2021). Nonetheless, there are good practices to improve feedback quality also at a larger scale. For instance, specifying required qualifications (Smith-Renner et al., 2020), using multiple annotators for each sample (Lertvittayakumjorn, Specia, and Toni, 2020), and having an initial training phase (Egelman, Chi, and Dow, 2014) are common strategies applicable prior to the feedback experiments.

AdaTest (M. T. Ribeiro and S. Lundberg, 2022) is particularly interesting as it addresses limitations in HitL caused by the high variability in human creativity to imagine and interpret bugs as well as the extensive labor necessary to fix them. Their work leverages LLMs like GPT-3 (Brown et al., 2020) to automatically write unit tests highlighting model bugs which are then fixed through an iterative test loop with humans. The authors test AdaTest on eight different NLP tasks and claim it is 5-10x more effective than previous methods.

2.6 XAI FOR MODEL ROBUSTNESS

The increasing adoption of NLP models in real-world scenarios necessitates robustness against adversarial text attacks, a requirement underscored by current legislative guidelines (European Commission, 2020). However, the literature shows that even the latest models remain vulnerable to input manipulations, which can mislead them effectively (Belinkov and Glass, 2019; W. Wang et al., 2019; W. E. Zhang et al., 2020). Failure at handling attacks hinders the safe deployment of such systems and has a detrimental impact on user trust and progress in the field (Xuezhi Wang, H. Wang, and Yang, 2022).

Although not yet formally understood, there is a clear connection between robustness and interpretability (Bhatt et al., 2020). First of all, adversarial samples are a widely used form of explanation to inspect for model weaknesses (see 2.3) and also share common traits with counterfactuals. At the same time, models which are more robust tend to produce better explanations (Etmann et al., 2019). For instance, Tsipras et al. (2019) show that enhancing model robustness via *adversarial training* leads to more accurate feature saliency maps.

Measuring feature attribution scores is also of great guidance when searching for the most effective adversarial perturbation (Ren et al., 2019). It feels quite intuitive as the features on which the model relies the most are the most profitable to target when trying to produce an incorrect prediction. Explanations carry strong signals for creating and thus identifying adversarial attacks. This is also confirmed by several works (Fidel, Bitton, and Shabtai, 2020; Tao et al., 2018; Ye et al., 2020) which leverage model explanations to detect manipulated inputs in computer vision. More in detail, Fidel, Bitton, and Shabtai (2020) uses SHAP signatures—i.e. SHAP features attribution explanations—from input images and feeds them to an ad-hoc classifier. Tao et al. (2018) improve robustness in face detection models by utilizing attributions at the neuron level to identify and amplify neurons critical for recognizing interpretable facial attributes. Finally, Ye et al. (2020) leverage saliency maps to recognize when the image classifier is focusing on unusual regions of the input and thus the input is likely manipulated. The authors test their approach successfully on popular object classification benchmarks such as ImageNet (J. Deng et al., 2009)

This hidden yet clearly existent relationship between adversarial attacks and model explanations is a substantial inspiration for two of the studies presented in this work (see 3). Indeed, both Mosca, Agarwal, et al. (2022) and Huber et al. (2022) further explore the robustness-interpretability connection in NLP and show that explanations in NLP also carry strong signals for adversarial detection.

3

PRESENTED STUDIES

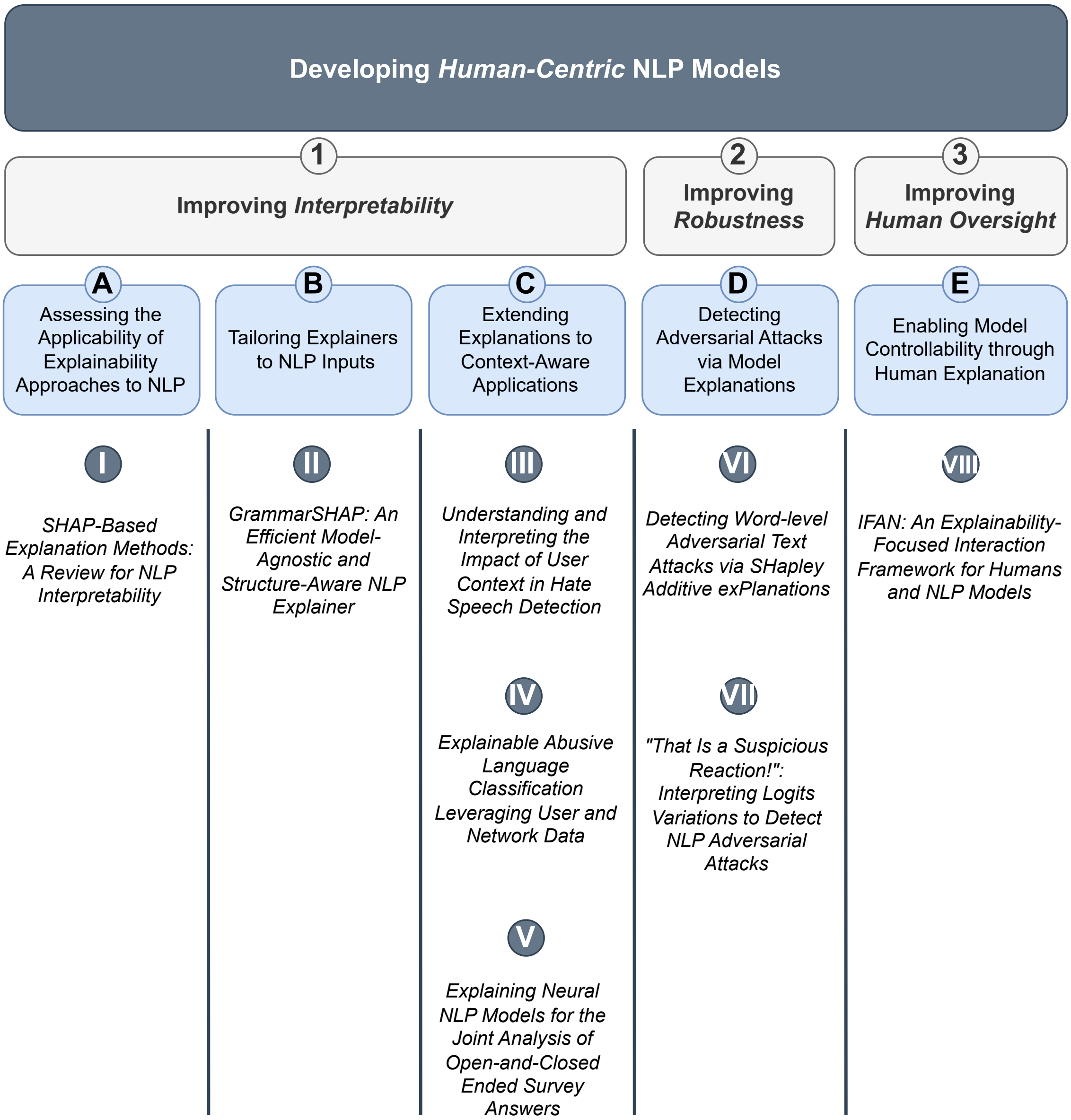

Developing *Human-Centric* NLP Models
1
Improving *Interpretability*
2
Improving *Robustness*
3
Improving *Human Oversight*
A
Assessing the Applicability of Explainability Approaches to NLP
B
Tailoring Explainers to NLP Inputs
C
Extending Explanations to Context-Aware Applications
D
Detecting Adversarial Attacks via Model Explanations
E
Enabling Model Controllability through Human Explanation
I
SHAP-Based Explanation Methods: A Review for NLP Interpretability
II
GrammarSHAP: An Efficient Model-Agnostic and Structure-Aware NLP Explainer
III
Understanding and Interpreting the Impact of User Context in Hate Speech Detection
IV
Explainable Abusive Language Classification Leveraging User and Network Data
V
Explaining Neural NLP Models for the Joint Analysis of Open-and-Closed Ended Survey Answers
VI
Detecting Word-level Adversarial Text Attacks via SHapley Additive exPlanations
VII
"That Is a Suspicious Reaction!": Interpreting Logits Variations to Detect NLP Adversarial Attacks
VIII
IFAN: An Explainability-Focused Interaction Framework for Humans and NLP Models

Figure 3.1: An overview of the conducted studies, together with how they address this work's research objectives.

This work presents eight scientific studies. All of them have been conducted within the time window (Aug. 2020 to Mar. 2023) of Edoardo Mosca's doctoral degree and work towards the objectives defined in section 1.3. Once again, we explore the usage of post-hoc explanations to improve the development of NLP models w.r.t. three aspects: *interpretability*, *robustness*, and *controllability* (also known as *human oversight*).

Figure 3.1 extends Figure 1.1 and connects each study to the research objectives it addresses. Based on the same structure, the studies are divided and presented accordingly in the following sections.

3.1 ASSESSING THE APPLICABILITY OF EXPLAINABILITY APPROACHES TO NLP

3.1.1 *Motivation*

Research efforts in interpretability extend well beyond the field of NLP (Bhatt et al., 2020; Madsen, Reddy, and Chandar, 2022). Plenty of post-hoc explainability approaches can be found in other application fields and for different data types—e.g. images, speech, graphs, tabular data, and time series (Guidotti et al., 2019; Nagahisarchoghaei et al., 2023). In practice, domain-agnostic frameworks often become the most widely utilized to produce explanations.

Popular frameworks such as *SHapley Additive exPlanations* (SHAP) (S. M. Lundberg and Lee, 2017), LIME (M. T. Ribeiro, S. Singh, and Guestrin, 2016), and IG (Sundararajan, Taly, and Yan, 2017) have received significant attention also for NLP applications. The SHAP framework in particular has become a gold standard for local explanations thanks to its solid theoretical foundation and broad applicability across different types of models.

Nevertheless, domain-agnostic frameworks come with their sets of challenges and limitations when applied to NLP. The unique nature of text data, with its inherent contextual information and complex structure, cannot be properly exploited by frameworks with a rather general-purpose design.

It becomes crucial to understand such limitations and investigate how subsequent research has sought to overcome them. This not only sheds light on the shortcomings of current approaches but also delivers a practical utility to identifying less-known and potentially more NLP-tailored interpretability methods.

3.1.2 *Study* I •

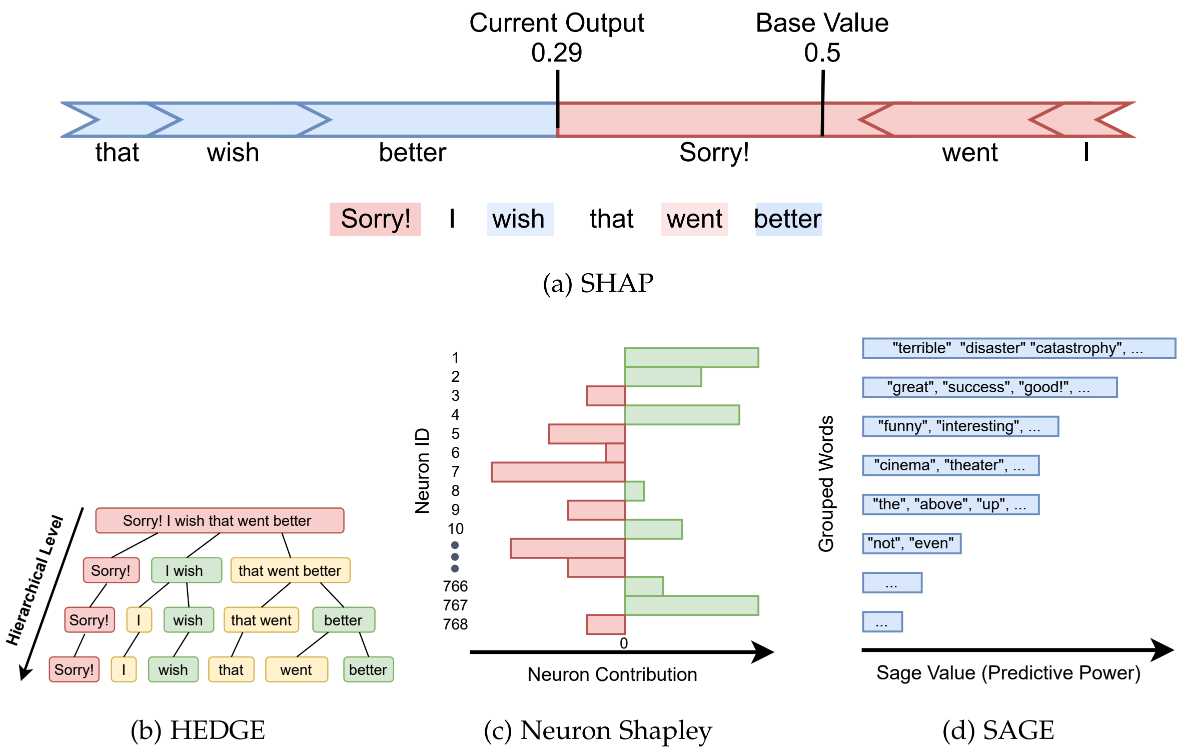
In our study "SHAP-Based Explanation Methods: A Review for NLP Interpretability" (Mosca, Szigeti, et al., 2022), we thoroughly examine and review the SHAP framework (S. M. Lundberg and Lee, 2017) and its derived variants in the context of NLP. After providing a concise summary of the necessary background around Shapley values (Shapley, 1953) and techniques for their estimations, our work's contribution is organized around three principal objectives:

(1) We identify five significant research streams that have emerged from the SHAP framework (S. M. Lundberg and Lee, 2017). These different yet overlapping directions categorize approaches working towards *tailoring explanations to different input data*, *explaining specific models*, *improving the framework's flexibility via modifying core assumptions*, *producing different explanation types*, and *estimating Shapley values more efficiently*.

(2) We conduct a detailed review of 41 distinct SHAP- and Shapley-value-based methods, each of which falls within one or more of the categories mentioned in the previous point. This review encompasses an assessment of the unique assumptions, input/model prerequisites, forms of explanation, and existing implementations for each method.

(3) Lastly, and crucially, we assess the suitability of each approach for NLP models and tasks. Approaches are ranked in four tiers ranging from *ready off-the-shelf* to *not relevant*. Based on the assessment's results, we also provide use-case-based recommendations and instructive examples for NLP researchers.

In an effort to maximize the practicality of our study for both practitioners and newcomers to the fields of XAI and NLP, we compile our findings, evaluations, and



(a) SHAP

(b) HEDGE

(c) Neuron Shapley

(d) SAGE

Figure 3.2: Instructive explanation examples provided in Mosca, Szigeti, et al. (2022, Figure 2-5) for sentiment analysis.

classifications of the 41 reviewed methodologies into a concise yet comprehensive overview table (Mosca, Szigeti, et al., 2022, Table 1). This table serves as a reference tool that encapsulates the essence of our research contribution, offering a quick yet holistic perspective on the diverse SHAP-based interpretability methods.

In the same spirit, we add visual examples throughout the work. Indeed, such visualizations—collected in Figure 3.2—sketch prototypical explanation outputs from the main reviewed approaches and help the reader to develop a practical understanding in terms of what kind of information output to expect when applying a certain method.

Subfigure (a) depicts a KernelSHAP-generated explanation, where each word contributes to the overall sentiment prediction, justifying the deviation from the model's average prediction (base value) to the current output (S. M. Lundberg and Lee, 2017). Subfigure (b), instead, shows a hierarchical explanation from HEDGE (H. Chen, Zheng, and Y. Ji, 2020), where Shapley values—negative (red), neutral (yellow), and positive (green)—are calculated for different levels of feature granularity.

Subfigure (c) sketches a Neuron Shapley (Ghorbani and J. Y. Zou, 2020) explanation for the neurons belonging to a BERT output layer (Devlin et al., 2019). Analogous to SHAP for input features, each neuron is assigned a Shapley value indicating their contribution towards or against the prediction. Lastly, Subfigure (d) illustrates an example of a SAGE explanation (Covert, S. M. Lundberg, and Lee, 2020), where the bars' length measures the global predictive power of each feature group, i.e. how useful those features are for the model's performance.

Study I's review serves as a beneficial guide, helping practitioners and researchers to navigate the landscape of SHAP-based interpretability and make informed decisions about which method is most adequate for their specific NLP use case. Furthermore, the analysis uncovers less-known SHAP-based approaches that have the potential to offer a more suitable solution for the target application. The full study can be found in Appendix A.1.

3.1.3 *Discussion*

Our work reviews a total of 41 SHAP-based methods for NLP interpretability. While this high number underscores the popularity and wide-ranging applicability of the SHAP framework (S. M. Lundberg and Lee, 2017), it also necessitated filtering steps to make the literature search process more manageable (Mosca, Szigeti, et al., 2022, Section 3). As expected, most of the methods fell into more than one of the five identified categories, reflecting the overlapping nature of these research streams.

The included examples and the overview table have been well received by fellow researchers and peer reviewers for the inherent practical utility they provide. Nonetheless, our review also highlighted some limitations of employing Shapley values in general (I. E. Kumar et al., 2020; Merrick and Taly, 2020; Sundararajan and Najmi, 2020). Moreover, while we were able to provide meaningful recommendations for a variety of NLP tasks, we acknowledge that SHAP-based methods—and feature attribution approaches in general—may not be the best option for some sequence-to-sequence tasks, which are becoming more prevalent in recent times.

3.2 TAILORING EXPLAINERS TO NLP INPUTS

3.2.1 *Motivation*

Feature attribution explanations have emerged as one of the predominant explainability tools (Arrieta et al., 2020; Bhatt et al., 2020). They highlight input components that significantly influence the resulting output. Yet, existing methods have predominantly focused on attributing relevance scores to individual words (Madsen, Reddy, and Chandar, 2022).

Treating words independently overlooks the context-dependency and structured nature of natural language (Mosca, Demirtürk, et al., 2022). In many cases, a word's meaning and carried sentiment can be dramatically reshaped by its surrounding context, position within a sentence, and relationship to other words. This highlights a gap in explainability research and motivates the need for explainers able to account for the sentence's structure and the words' interdependencies.

Some recent works focus on phrase-level and hierarchical explanations to address such limitations. Techniques to identify structure and dependencies include exhaustive search (Tsang et al., 2018), combining contextual decomposition scores (C. Singh, W James Murdoch, and Yu, 2018), using Shapley interactions or Bahnhaf values from predefined tree structures (J. Chen and Jordan, 2020; S. M. Lundberg, G. G. Erion, and Lee, 2018), and breaking down text iteratively based on the (directly detected) weakest words' interaction (H. Chen, Zheng, and Y. Ji, 2020).

Nonetheless, there remains a pressing need for further research and development of methods that can fully capture the complexity and nuances of natural language. Approaches can also be built as extensions of existing popular frameworks, thereby leveraging the properties and advantages for which they are renowned.

3.2.2 *Study* II †

In our study "GrammarSHAP: An Efficient Model-Agnostic and Structure-Aware NLP Explainer" (Mosca, Demirtürk, et al., 2022), we work towards explanations that align with the input instance's linguistic structure. We do so by building hierarchical explanations which attribute relevance scores to sentence constituents across multiple levels. In contrast with preceding studies tackling similar issues (J. Chen and Jordan, 2020; Tsang et al., 2018), we directly extend the SHAP framework and take advantage of the theory backing it (S. M. Lundberg and Lee, 2017).

Figure 3.3 sketches an overview of our study's proposed methodology, whose primary contributions include:

(1) We propose GrammarSHAP, a model-agnostic hierarchical explainer accounting for the sentence constituents and their interdependencies. In particular, we couple a constituency parsing layer for merging multi-word tokens with a custom KernelSHAP adapted for efficiency at run-time.

(2) We advocate for the removal of the standard SHAP background dataset, opting instead to utilize masking tokens. This modification speeds up the pipeline and reduces unwanted explanation anomalies.

(3) We perform a qualitative comparison of our technique with existing ones, focusing on the quality of explanations produced and the computational resources required.

GrammarSHAP leverages the Berkeley Neural Parser (Kitaev and Klein, 2018) to iteratively merge multi-word tokens and reflect the sentence grammar's structure. Starting from the single-word level ($depth = 0$) and proceeding until the entire sentence is a single token ($depth = N$), we can retrieve word groups and constituents at any depth. Our implementation resolves inconsistencies between BERT's sub-word tokens for OOV words and Berkeley Parser's requirement for full words.

To retain model agnosticism, we pick KernelSHAP from the SHAP framework as a baseline to build upon (S. M. Lundberg and Lee, 2017). Directly extending to

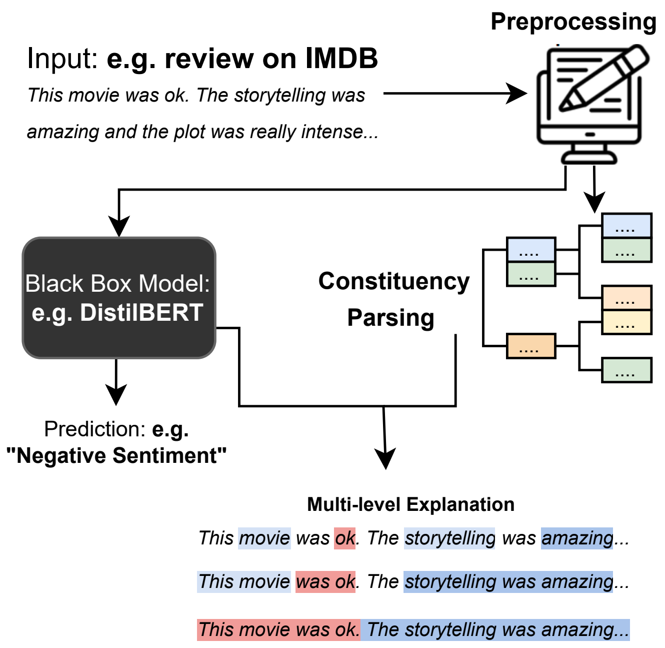
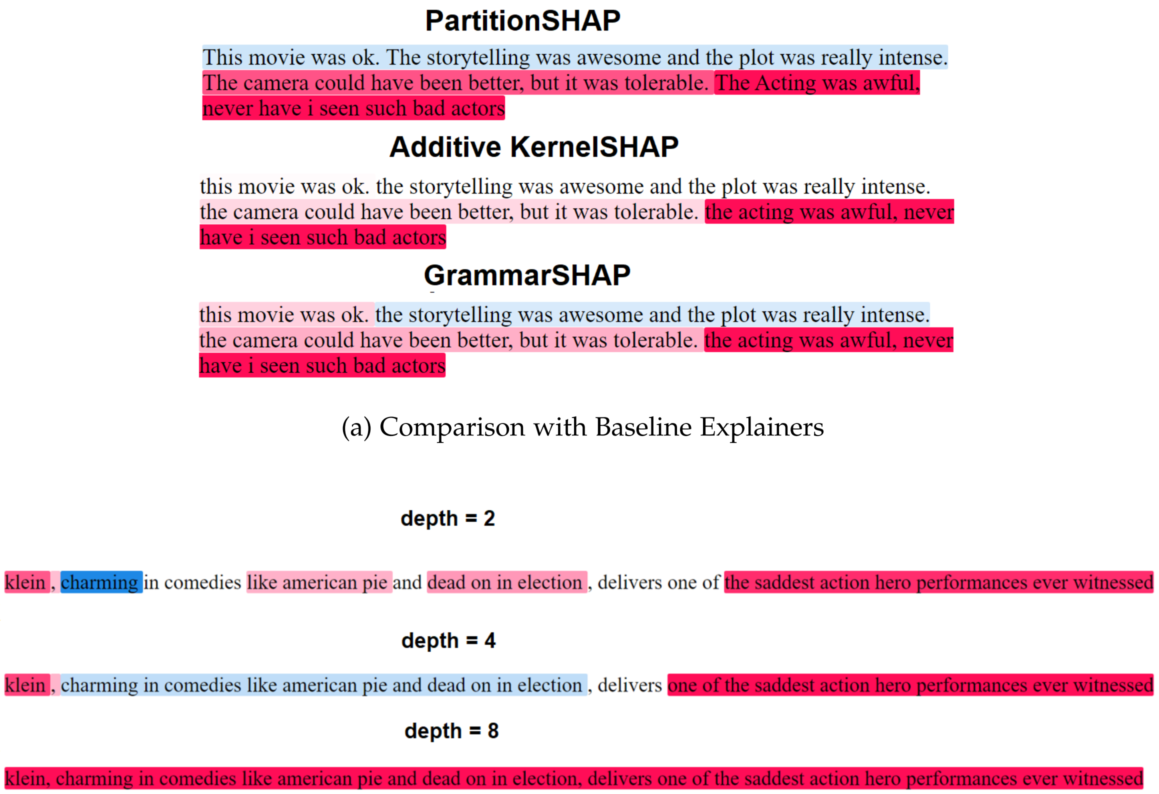

Preprocessing
Input: e.g. review on IMDB
This movie was ok. The storytelling was amazing and the plot was really intense...
Black Box Model: e.g. DistilBERT
Constituency Parsing
Prediction: e.g. "Negative Sentiment"
Multi-level Explanation
This movie was ok. The storytelling was amazing...
This movie was ok. The storytelling was amazing...
This movie was ok. The storytelling was amazing...

Figure 3.3: Overview of GrammarSHAP's pipeline (Mosca, Demirtürk, et al., 2022, Figure 1). The input instance is parsed in its constituents in parallel to the model prediction. These are used as a base for multi-token masking and thus enable creating feature attribution scores at multiple granularity levels.

multi-word scores—and thus multi-word input perturbations—leads to challenges such as (i) computational inefficiency, (ii) unidirectional explanations, and (iii) high attributions for [SEP] tokens due to its alteration of sentence length when used as a substitute from the background data (Mosca, Demirtürk, et al., 2022). We mitigate these issues by replacing sampling from background samples with simple [MASK] tokens, which accelerates the explainer process by $\sim$ 60x and eliminates [SEP] related explanation artifacts as it is no longer part of the background data.

Figure 3.4 showcases qualitative comparisons for explanations generated for a sentiment analysis task. Specifically, the first image compares explanations generated by GrammarSHAP and other baseline explainers, while the second showcases GrammarSHAP's explanations obtained at different hierarchical levels.

GrammarSHAP is proficient in identifying both positive and negative contributions at various granularity levels, also where other explainers struggle to do so (Mosca, Demirtürk, et al., 2022). Nevertheless, despite delivering a substantial speed



PartitionSHAP

This movie was ok. The storytelling was awesome and the plot was really intense. The camera could have been better, but it was tolerable. The Acting was awful, never have i seen such bad actors

Additive KernelSHAP

this movie was ok. the storytelling was awesome and the plot was really intense. the camera could have been better, but it was tolerable. the acting was awful, never have i seen such bad actors

GrammarSHAP

this movie was ok. the storytelling was awesome and the plot was really intense. the camera could have been better, but it was tolerable. the acting was awful, never have i seen such bad actors

(a) Comparison with Baseline Explainers

depth = 2

klein , charming in comedies like american pie and dead on in election , delivers one of the saddest action hero performances ever witnessed

depth = 4

klein , charming in comedies like american pie and dead on in election , delivers one of the saddest action hero performances ever witnessed

depth = 8

klein, charming in comedies like american pie and dead on in election, delivers one of the saddest action hero performances ever witnessed

(b) Output at different hierarchical levels

Figure 3.4: Explanation comparison between (a) three explainers for grouped features relevance (5th level) (Mosca, Demirtürk, et al., 2022, Figure 3) and (b) three GrammarSHAP's outputs for different hierarchical levels—2nd, 4th, and 8th.

up compared to trivially extending KernelSHAP (additive KernelSHAP), it is still considerably slower than PartitionSHAP (S. M. Lundberg and Lee, 2017).

The full study can be found in Appendix B.1.

3.2.3 *Discussion*

Study II introduces GrammarSHAP, extending the SHAP framework to build hierarchical explanations that meaningfully reflect the sentence structure. In line with the findings from the related work, we observe the challenges raised by considering interactions between words and calculating the importance of multi-word tokens due to the combinatorial explosion it presents (H. Chen, Zheng, and Y. Ji, 2020; S. M. Lundberg and Lee, 2017).

While our adaptations mitigate such issues and considerably speed up the execution time, the explainer may prove to be too slow for applications involving particularly long texts. Still, we maintain that GrammarSHAP is efficient considering the granularity of feature contributions it can detect (Mosca, Demirtürk, et al., 2022).

When assessing the explanations' quality, our evaluation process revolves around the introduced methodological strategies along with a qualitative analysis of the produced explanations. Although evaluation metrics for explanations are difficult to establish and lack standardization as of now (see 2.4), future work should set quantitative diagnostics as a priority for their comparison (Atanasova et al., 2020).

3.3 EXTENDING EXPLANATIONS TO CONTEXT-AWARE APPLICATIONS

3.3.1 *Motivation*

In many real-world NLP use cases, models solely dependent on textual data may encounter limitations in both performance and generalizability—often neglecting the social context in which a natural language utterance was produced. This is especially true in applications where context is paramount, such as hate speech detection (Mishra et al., 2018), opinion mining (Sundermann et al., 2019), and other tasks probing personal traits, intentions, and opinions (Gencheva et al., 2019).

In these scenarios, NLP models exploiting context through multiple input modes show notable success compared to their text-based counterparts (Fehn Unsvåg and Gambäck, 2018). However, a simple performance comparison does not fully reveal the implications of incorporating additional context and user information into these models.

Employing explainability provides an opportunity to extract deeper insights into how different data sources interact and hence how context shapes the deployed models' workings. Especially when different post-hoc approaches are used complementarily, we can gain a holistic perspective on the trained model, which becomes an additional source of knowledge and further enhances our understanding of the data.

3.3.2 *Study* III •

Our study "Understanding and Interpreting the Impact of User Context in Hate Speech Detection" (Mosca, Wich, and Groh, 2021) leverages post-hoc explainability to investigate the effect that contextual features have on hate speech classifiers.

Previous works were able to improve detection performance by adding *user features* such as the gender (Waseem, 2016), geolocation (Galán-Garcıa et al., 2016), and the number of followers/friends (Fehn Unsvåg and Gambäck, 2018) as well as modeling *online interactions and relationship* (Mishra et al., 2018, 2019; M. H. Ribeiro et al., 2018). Yet, beyond accuracy, little attention has been given to the additional changes that such features bring to the models.

Study III demonstrates that including such features can significantly alter the behavior and characteristics of the recognition algorithms (Mosca, Wich, and Groh, 2021). Its contribution can be summarized as follows:

(1) Our study validates that incorporating user and social context into the models is indeed the reason for performance gains. Concurrently, we examine the feature space learned by the models to comprehend how these additional features are leveraged for detection purposes.

(2) We find that models with context exhibit reduced bias from the text itself. However, such incorporation unfortunately introduces new forms of bias, which is characterized in our explainability analysis.

The experimentation focuses on capturing the behavioral differences between two simple detection models, one that solely relies on text features—i.e. *text model*—and one that instead also incorporates context features—i.e. *social model*. For training and testing, we choose to utilize two popular Twitter-based datasets from Waseem (2016)

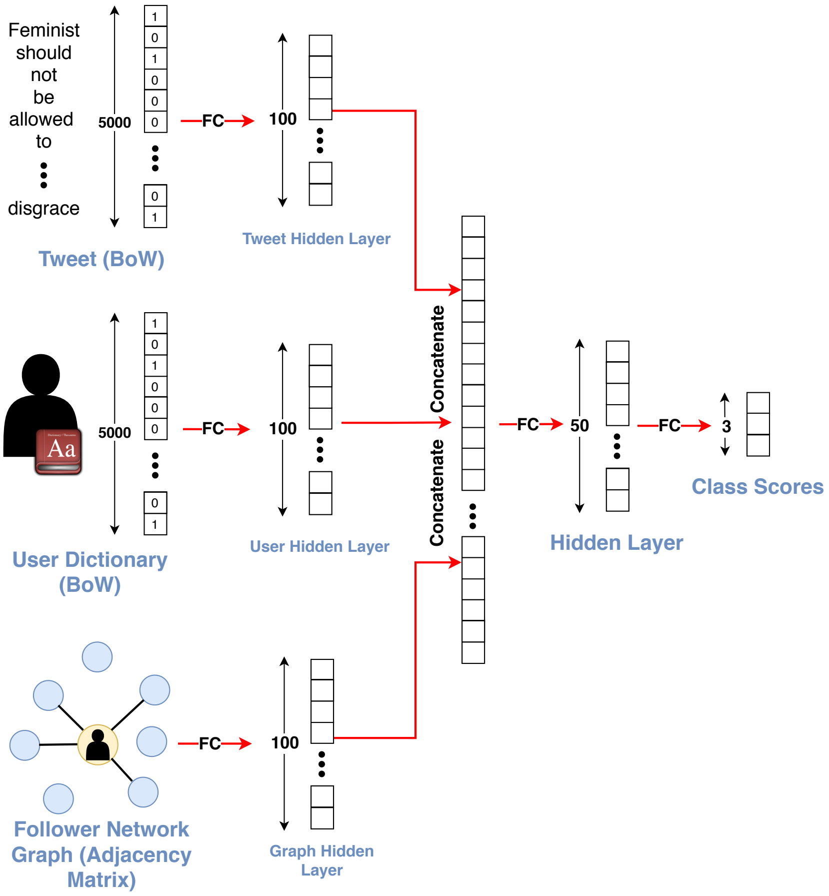


Figure 3.5: Architectures for the text model (top branch only) and social model (all branches) (Mosca, Wich, and Groh, 2021, Figure 1 and 2). The three input streams are initially processed separately and then their intermediate representations from the different branches are concatenated together and fed to two more layers to compute the output probabilities.

and Davidson et al. (2017) due to their diverse speech categories and widespread use as detection benchmarks. Both datasets have three classes—*racism/sexism/none* and *hate/offensive/none* respectively—and our models achieve a satisfactory performance (Mosca, Wich, and Groh, 2021, Table 1 and 2). For each dataset, based on the anonymized user metadata provided, the additional context features are then retrieved via the Twitter API.

Figure 3.5 sketches the detection architectures, including the three input sources and how they are processed together. The text model is only composed of the top branch, i.e. the tweet's content itself. The social model, on the other hand, learns also from two additional modes—the user's overall language use and their follower network. The former is obtained by combining the bag-of-words representations from all their tweets. The latter—similarly to prior state-of-the-art hate speech detectors (Mishra et al., 2018, 2019)—is sourced from the adjacency matrix of the retrieved community graph.

Results stemming from feature attribution via Shapley values and embedding of artificially crafted tweets into the model's learned embedding space (Mosca, Wich, and Groh, 2021, Figure 3-6) underline the different behavior that the two models present. The integrated context—especially the user vocabulary—plays a key role in the social model's performance gains, otherwise not justifiable by architectural differences with the text model (Mosca, Wich, and Groh, 2021, Figure 3). Moreover, the social model learns a cluster-like landscape based on user traits that simplifies the classification process (Mosca, Wich, and Groh, 2021, Figure 4).

While the text model's prediction can be easily altered by changing the hate target, the social model demonstrates greater resilience to text manipulations. On the other hand, swapping the tweet's author provokes unwanted output changes for the social model (Mosca, Wich, and Groh, 2021, Figure 5-6). Hence, although user-derived features can reduce text bias, they can also introduce a new bias that excessively discriminates against users based on past behavior, thus complicating accurate hate content classification.

The full study can be found in Appendix A.2.

3.3.3 *Study* IV †

In the same spirit as study III, our work "Explainable Abusive Language Classification Leveraging User and Network Data" (Wich, Mosca, et al., 2021) identifies limitations in utilizing models purely based on text. Hence, it further explores extending detection pipelines to also leverage user and network data as well as post-hoc explainability for multi-modal models and inputs.

More in detail, the contribution of study IV can be summarized as follows:

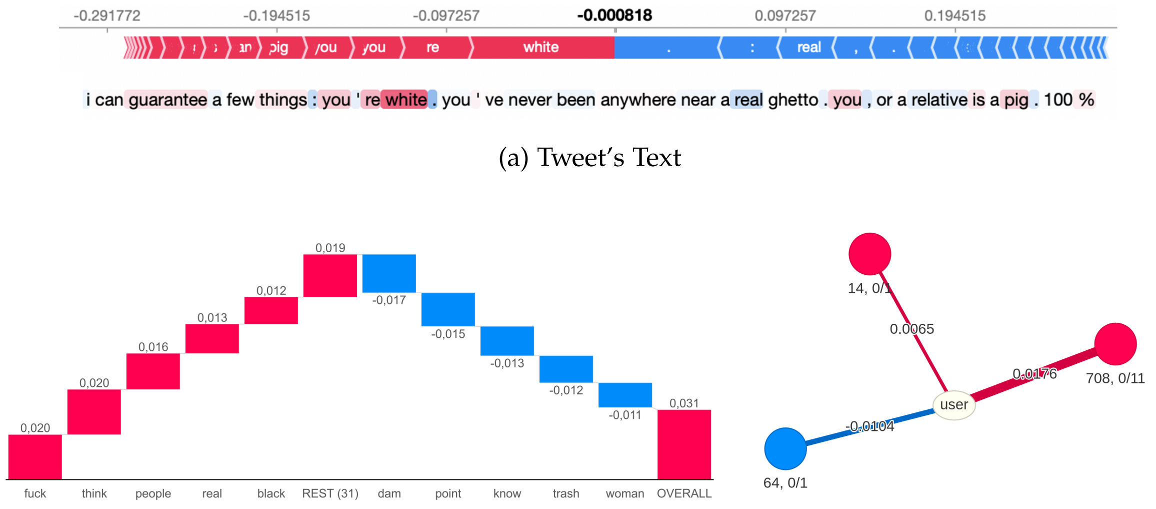
(1) We propose a state-of-the-art hate speech detector composed of three input modes, each handled by a different sub-model:

  - *Text Model:* Uses DistilBERT (Sanh et al., 2019) with a classification head to process the tweet text meant for classification. Usernames are stripped from tweets prior to tokenization to prevent classifier bias.
  - *History Model:* Implements bag-of-words to model the user's tweet history, reflecting the 500 top dataset terms based on TF-IDF appearing in the user's tweets.
  - *Network Model:* Utilizes the GraphSAGE (Hamilton, Ying, and Leskovec, 2017) inductive representation learning framework to model the user's social network. By training on the undirected network graph of social relations, it is able to generate embedding for new users.

(2) We extend SHAP explanations (S. M. Lundberg and Lee, 2017) to operate also with multi-modal inputs. In particular, we can produce visualizations based on Shapley values for the text, user history, and network streams.

Our multimodal model is trained and evaluated on three datasets—those provided by Davidson et al. (2017), Waseem (2016), and Wich, Breitinger, et al. (2021). Results from our ablation study (Wich, Mosca, et al., 2021, Table 3) indicate once more that user and network data enhance abusive language detection. Nonetheless, the leap in F1 score performance is somewhat more modest compared to study III (Mosca, Wich, and Groh, 2021, Table 1) and ranges from 0.1pp to 2.4pp. This modest improvement can be attributed to two factors. Firstly, leveraging DistilBERT for text classification provides a stronger baseline compared to the one used in Mosca, Wich, and Groh (2021). Second, the network data of the datasets is noticeably sparse as their collection was not performed on a connected subnetwork of users (Wich, Mosca, et al., 2021).

Concerning explainability, the insights extracted reveal the individual roles of each submodel and contribute significantly to the comprehensibility of the prediction for human understanding. Figure 3.6 illustrates an example of a tweet incorrectly labeled as neutral by the text submodel (weights of neutral words, marked in blue, surpasses the score of other groups, in red). However, this misclassification was rectified by the



(a) Tweet's Text

(b) User's History

(c) User's Network (each node representing a community)

Figure 3.6: Explanations produced by the (a) text, (b) history, and (c) network submodels (Wich, Mosca, et al., 2021, Figure 4). These are presented as Shapley Values. Red—i.e. positive values—favors a classification as abusive. Blue—i.e. negative values—supports favor a classification as non-hateful.

other two submodels, whose explanations show red (=abusive) features prevailing in importance.

The full study can be found in Appendix B.2.

3.3.4 *Study* V •

Our study "Explaining Neural NLP Models for the Joint Analysis of Open- and Closed-Ended Survey Answers" (Mosca, Hermann, et al., 2022) investigates using NLP transformers in conjunction with post-hoc explainability to automatically extract knowledge and interpret correlations in large-scale surveys' answers.

Including both open- and closed-ended questions is challenging as models have to simultaneously deal with structured and unstructured data. Given the limitations of the currently dominant practices (Eichstaedt et al., 2021)—heavily based on human labor or shallow NLP pipelines—our work contributes the following:

(1) We employ the widely-used DistilBERT (Sanh et al., 2019) transformer model to tackle open-ended queries, surpassing the precision of traditional methods in capturing contextual correlations in the text.

(2) We apply several variants of SHAP (S. M. Lundberg and Lee, 2017) to examine both instance-level feature importance as well as high-level concepts learned by the model (Yeh et al., 2020). Combining such approaches at several granularity levels contributes to a holistic understanding of what our model has learned.

(3) Our methodology delivers promising results on the EMS 1.0 dataset—which explores factors impacting students' career aspirations (Gilmartin et al., 2017). Indeed, it identifies and reveals relevant factors from both closed-ended and open-ended text responses.

To automatically analyze a survey with an NLP pipeline, we convert the structured questions/answers format into a predictive task. For instance, for the case of the EMS 1.0 dataset, we utilize answers regarding the student's contextual situation—e.g. *background*, *learning experiences*, *current influences and values*—as input to predict their answers about career goal aspirations—i.e. scores regarding working in a *startup*, *large company*, *university*, etc. (Gilmartin et al., 2017; Mosca, Hermann, et al., 2022, Appendix A).

Figure 3.7 depicts the overall model architecture. We test various alternatives to process the text coming from open-ended answers (left) and pick *DistilBERT + mean* as the best-performing choice (Mosca, Hermann, et al., 2022, Table 2). Closed-ended answers are instead fed to fully connected layers (right) and then processed jointly with the extracted text contextual representation. Our performance experiments and ablation study across all output objectives show our models using both text answers (however, not all) and numerical inputs to achieve the highest scores (Mosca, Hermann, et al., 2022, Table 1).

Concerning interpretability, we combine low-level features and neuron explanations with high-level concept ones. For the former, we (1) compute and compare SHAP values for both textual and numerical value embeddings' neurons, (2) extract text segments triggering the neurons with the highest activation, and (3) determine input

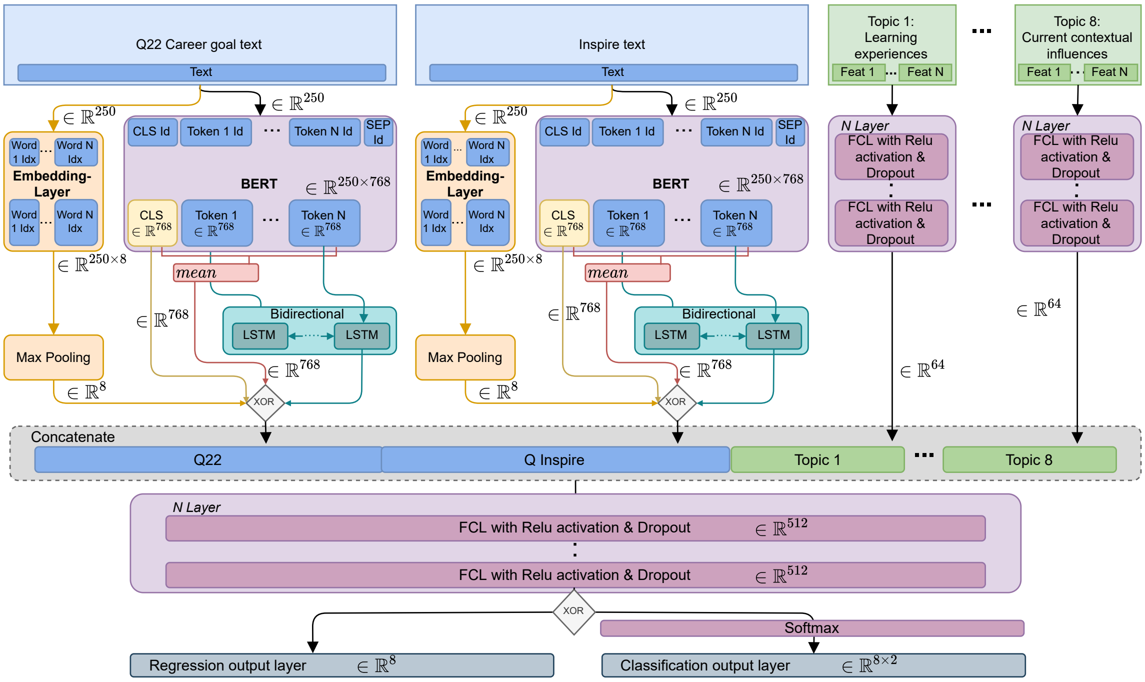


Figure 3.7: Multi-modal model classification architecture—combining both text and numerical (i.e. categorical) features (Mosca, Hermann, et al., 2022, Figure 1). XOR symbols denote the various options tested for different sub-component models.

SHAP values w.r.t. model output. Concerning the latter, we extract concepts via ConceptSHAP (Yeh et al., 2020) to capture how the model organizes higher-level information. After extraction, we (1) describe each concept via its K nearest neighbors, (2) measure the influence of each concept for single predictions, and (3) report *completeness scores* - i.e. how well the set of extracted concepts describes the model's behavior (Yeh et al., 2020).

Results show that low-level explanations offer key insights about overall question relevance and student-specific factors that influence their entrepreneurial aspirations. The automated high-level analysis instead identifies relevant concepts—such as *clarity of career plans*, *career characteristics*, and *plan timeline*—that are in line with earlier research based on human judgment (Grau et al., 2016).

The full study can be found in Appendix A.3.

3.3.5 *Discussion*

The centrality of context is a recurrent theme for many NLP use cases as models often operate in an ecosystem rather than in isolation. Integrating the surrounding context of a specific text instance can be pivotal for accurate predictions and conclusions. On the same line, findings from studies III, IV, and V consistently demonstrate the performance advantages of utilizing models leveraging context through multiple input modes.

Each of these studies works on combining and occasionally expanding existing post-hoc explainability frameworks. The goal is to build a holistic understanding of the trained models and hence exploit them as an additional resource for insights and knowledge about the data.

Study III and IV work on hate speech detection. The first work shows the insufficiency of performance metrics for the sake of model comparison. As an intriguing insight from explanations, incorporating context helps to counterbalance biases found within the text, but can potentially introduce novel forms of bias derived from the context itself. The second, instead, carries out further ablation studies for a context-aware model and extends explanations to also function on the different input modes.

Finally, study V works on survey answers and looks at how explainability can serve us when models have to deal with both structured (i.e. multiple-choice answers) and unstructured data (text or open-ended answers). It showcases the potential of merging feature attribution with concept analysis, which ultimately facilitates the automated analysis of survey—a process that would traditionally necessitate a considerable amount of human effort.

While all works contribute to the objective, it generally remains hard to find one-fits-all pipelines. Therefore, the introduced methodologies are often limited to the specific use cases they're designed for.

3.4 DETECTING ADVERSARIAL ATTACKS VIA MODEL EXPLANATIONS

3.4.1 *Motivation*

Adversarial attacks were discovered roughly a decade ago and are input samples artificially manipulated to trick the model into making a wrong prediction (Szegedy et al., 2014). To this day they remain effective against machine learning models despite advancements in architectures, data quality, and training methods (Yuan et al., 2019). NLP is unfortunately no exception to the rule (W. Wang et al., 2019; Xuezhi Wang, H. Wang, and Yang, 2022; W. E. Zhang et al., 2020).

Adversarial samples in NLP differ substantially from their computer vision counterpart. Thus, despite the large amount of study that has been carried out for attacks and defenses with images, most ideas cannot be directly reapplied to the text domain (Xuezhi Wang, H. Wang, and Yang, 2022). Most notably, images are continuous inputs that remain valid if they undergo a continuous perturbation. This is not the case for natural language, where the *discrete nature* of the text space limits the usage of gradient-based techniques and other continuous transformation (Lei et al., 2019). Furthermore, text inputs need to fulfill lexical, grammatical, and semantic constraints to properly convey meaning.

Numerous attacks manipulate samples at the character level and capitalize on *visual similarity*—e.g. DeepWordBug (J. Gao et al., 2018), HotFlip (Ebrahimi et al., 2018), and VIPER (Eger et al., 2019). However, they lead to non-existing terms and introduce syntactical inconsistencies (D. Pruthi, Dhingra, and Zachary C. Lipton, 2019; W. E. Zhang et al., 2020). Conversely, word-level attacks are effective at maintaining *semantic coherence* without noticeable discrepancies, thereby eluding spell check detection (Garg and Ramakrishnan, 2020; Ren et al., 2019).

Without effective defense strategies against adversarial instances, systems are likely to fail when attacked, thus jeopardizing safe model deployment and undermining public trust. In this regard, a fruitful strategy to tackle this challenge is *adversarial* *detection*—where the goal is not only to defend the model, but also to explicitly identify attacking attempts (Fidel, Bitton, and Shabtai, 2020; Mozes et al., 2021; Ye et al., 2020; Y. Zhou et al., 2019).

3.4.2 *Study* VI †

Our study "Detecting Word-Level Adversarial Text Attacks via SHapley Additive exPlanations" (Huber et al., 2022) investigates utilizing model explanations to detect manipulated text inputs explicitly. It builds on the intuition that—even if adversarial and original inputs look indistinguishable—the model still reacts differently to them and this can be captured by explainability (Fidel, Bitton, and Shabtai, 2020; Mosca, Agarwal, et al., 2022).

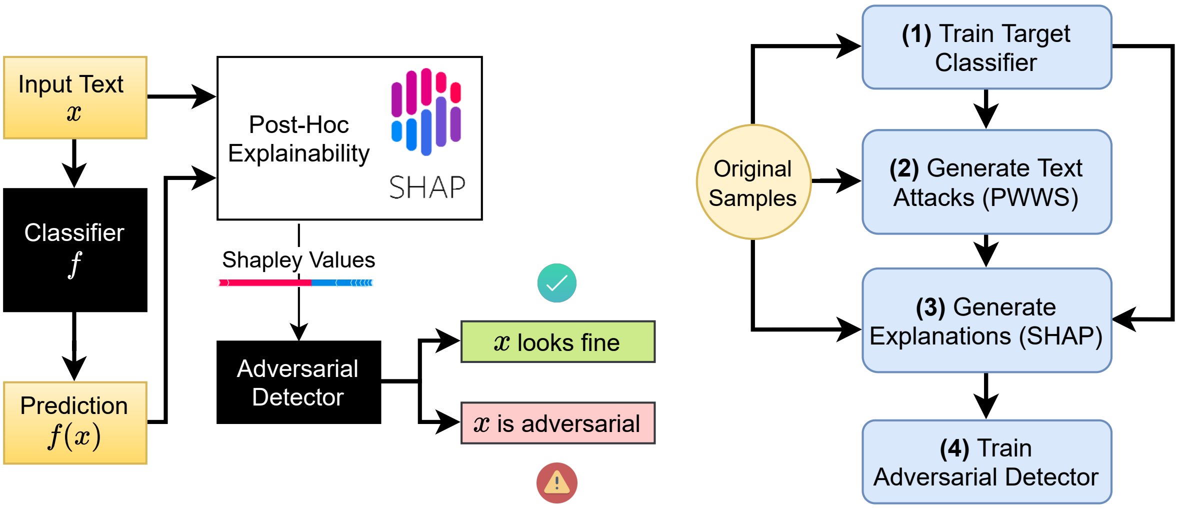
Our work draws inspiration from an analogous idea from computer vision (Fidel, Bitton, and Shabtai, 2020) and contributes the following:

(1) We propose an adversarial detector harnessing SHAP (S. M. Lundberg and Lee, 2017) to recognize text attacks. The approach outperforms the previous state of the art (Mozes et al., 2021; Y. Zhou et al., 2019) on four datasets.

(2) We evaluate our method w.r.t. data efficienty and generalization capabilities. It continues to perform competitively with little training data and is comparable to the prior when tested on unseen datasets.

(3) Alongside quantitative experiments, we project the space of generated SHAP explanations to two dimensions via UMAP (McInnes, Healy, and Melville, 2020). The resulting visualization shows most explanations corresponding to attacks to be easily separable from the remaining samples, which sheds light on the reason behind the method's success.

Figure 3.8 summarizes the proposed methodology. Given an input instance x and a task-specific classifier f potentially targeted by text attacks, we take the input/output pair to compute a SHAP explanation (S. M. Lundberg and Lee, 2017). The resulting



(a) Goal Pipeline

(b) Construction Steps

Figure 3.8: Overview of (a) the overall pipeline once the adversarial detector is ready to be deployed with the target classifier and (b) the steps required to train it (Huber et al., 2022, Figure 1).

Shapley values are then fed to the detector, which predicts whether the input is adversarial or not.

To train the detector, we utilize the classifier to craft a large number of attacks with PWWS (Ren et al., 2019). Then, both original and adversarial samples are fed to SHAP to generate explanations and train the detector. Please note that (i) the training procedure is only necessary once and (ii) the approach is model-agnostic as there are no assumptions on f.

We evaluate our approach on PWWS-generated samples attacking a Bi-LSTM (Schuster and Paliwal, 1997) model on four datasets. A few simple detector architectures such as SVM and Random Forests were tested; all reported similar (and satisfactory) results (Huber et al., 2022, Table 1). Further experiments show that adversarial detectors learn to perform the discrimination tasks accurately with as little as $\sim 2,500$ samples (Huber et al., 2022, Figure 4). Generalization experiments—i.e. training and testing on attacks coming from different datasets—report promising results (Huber et al., 2022, Table 4). However, we can still observe a steep decline in performance when compared to settings with no dataset mismatch.

The full study can be found in Appendix B.3.

3.4.3 *Study* VII •

Following the path of study VI, our work " 'That Is a Suspicious Reaction!': Interpreting Logits Variation to Detect NLP Adversarial Attacks" (Mosca, Agarwal, et al., 2022) further investigates measuring the model's reaction to explicitly detect adversarial input instances. Furthermore, it also builds its methodology based on the success of prior work using logits-based explanations and metrics to discriminate manipulated images (Aigrain and Detyniecki, 2019; Hendrycks and Gimpel, 2016; Yaopeng Wang et al., 2021).

The study contributes as follows:

(1) We present a model-agnostic logits-based metric, termed *Word-level Differential Reaction* (WDR), identifying words having a suspiciously high influence on the prediction. Moreover, this metric is not dependent on the amount of output classes.

(2) Leveraging WDR scores, we develop an adversarial detector capable of differentiating between original and syntactically-correct adversarial text inputs. Our methodology significantly outperforms the current state of the art in NLP.

(3) We show WDR-based detectors to possess full transferability capabilities and to generalize across various datasets, attacks, and target models without any need for retraining. Our test settings encompass transformers as well as both contextual and genetic attack techniques.

(4) Our primary hypothesis is validated via post-hoc explainability. i.e., the detector identifies adversarial patterns through the WDR scores. As an interesting insight, only a small portion of these scores actually carry a strong signal for adversarial detection.

Semantically similar adversarial attacks substitute a small number of words to transform the output, implying that the replaced words significantly affect the output (Alzantot et al., 2018). The WDR metric describes the reaction of a target model f to a given input x. To elaborate, the effect of removing an input word x_i on the prediction is quantified by the formula:

$$WDR(x_i, f) = f(x \backslash x_i)_{y^*} - \max_{y \neq y^*} f(x \backslash x_i)_y$$

where y^* is the predicted class and $f(x \backslash x_i)_y$ indicates the output logit for class y when the input sample x is presented without the word x_i. Often, when x is adversarial, we can expect to find perturbed words to have a negative $WDR(x_i, f)$ as their removal should restore the original prediction.

We feed the list of all $WDR(x_i, f)$—i.e. one score for each word x_i—as inputs to a machine learning detector and label it as either *original* or *adversarial*. The training follows analogous steps to study VI: i.e. (i) generation of a large number of adversarial attacks, (ii) computation of WDR scores for both original and adversarial samples, and (III) detector training via feeding WDRs.

After initial experiments to pick the best performing detector architecture (Mosca, Agarwal, et al., 2022, Table 2)—i.e. XGBoost (T. Chen and Guestrin, 2016)—we evaluate our pipeline against the state of the art from Mozes et al. (2021). With a collection of 28 configurations, i.e. 28 *target model/dataset/attack* triplets, the detector is exclusively trained on a pair of these configurations and subsequently tested on the remaining ones without any fine-tuning or retraining. Besides four different datasets, these configurations include target models like BERT (Devlin et al., 2019), DistilBERT (Sanh et al., 2019), and LSTM (Hochreiter and Schmidhuber, 1997) as well as adversarial examples generated with greedy attacks (Ren et al., 2019, PWWS), contextual attacks (Garg and Ramakrishnan, 2020, BAE), and genetic algorithms (W. Wang et al., 2019, IGA).

We outperform Mozes et al. (2021) in 22 out of 28 configurations (equal in 1, worse in 5), with an overall average F1 score improvement of 8.96pp (Mosca, Agarwal, et al., 2022, Table 3). Additionally, our method demonstrates a high adversarial recall, indicating a minimal number of false negatives, i.e. undetected attacks. Concerning generalization, the detector performs well across several attacks and target models, with noticeable drops in performance only on one dataset. Notably, the baseline FGWS struggles against text attacks from BAE, able to introduce context-aware perturbations.

Our study additionally assesses different *decision threshold* choices for the detector. A lower threshold implies a more cautious approach, increasing the likelihood of identifying an input as adversarial. Reducing the threshold probability from the default of 0.5 to 0.15 can boost adversarial recall above 98% with only a minor loss in F1-score ($< 2\%$) (Mosca, Agarwal, et al., 2022, Table 4). This is beneficial in scenarios where overlooking attacks (false negatives) has severe implications and attacks are infrequent or false positives only have a minimal impact.

Plotting the detector's SHAP values in relation to WDR scores indicates that only the highest scores significantly influence the adversarial detector (Mosca, Agarwal, et al., 2022, Figure 3). This aligns with our initial assumption that only a few word replacements significantly alter output logits, making their variation a useful measure for detecting input manipulations.

The full study can be found in Appendix A.4.

3.4.4 *Discussion*

Word-level adversarial examples are particularly challenging as they can deceive the target model while preserving semantics and without introducing grammatical and lexical inconsistencies. Both studies VI and VII show that instance-level explanation signatures carry rich signals. These signals can be harnessed to detect word-level adversarial attacks at scale, employing model-agnostic methodologies. While the first study investigates the usage of Shapley value for this purpose, the second one introduces a custom metric measuring the model's reaction when specific words are removed from the input.

Study VI already shows promising performance and generalization results compared to the state of the art. Moreover, the study reveals that the explanations space—as opposed to the input space—is simpler to navigate when it comes to discriminating maliciously manipulated samples. However, the evaluation process only examined a limited number of configurations, which may not comprehensively represent real-world scenarios.

On the other hand, study VII expands on the previous work and scales up the number of experiments, which consider a wider range of target models and text attack methods. Performance results show the approach to significantly outperform the state of the art and demonstrate superior generalization capabilities.

Despite improvements, there are still limitations that are not solved by study VII. In fact, the detection approach primarily focuses on word-level attacks and may struggle against adaptive attacks and newer adversarial strategies operating at the sentence level. Such attacks might not need to rely on a few token replacements to induce an output change. At the same time, handling particularly long texts could lead to extended computation times to calculate WDR scores.

3.5 ENABLING MODEL CONTROLLABILITY THROUGH HUMAN EXPLANATIONS

3.5.1 *Motivation*

The opacity of LLMs—and more generally of complex NLP models—not only hampers our ability to interpret and understand their inner workings, but also limits the influence we can exert over them. *Human oversight* is an essential safeguard to create highly-performing models that also align with ethical goals and values.

Especially as models' capability and autonomy increase, it becomes vital to research and develop tools and practices to control models effectively. The recent literature in XAI and HitL has therefore seen increased research effort (Monarch, 2021; Z. J. Wang et al., 2021), also contributing toolkits and frameworks for analyzing and improving complex NLP models (P. Liu et al., 2021; E. Wallace, Tuyls, et al., 2019).

Some works also offer low-code interfaces for stakeholders with no technical proficiency. Popular examples are ExplainaBoard from P. Liu et al. (2021), an interactive leaderboard providing detailed diagnostics of NLP models, LIT from Tenney et al. (2020), an open-source platform that visualizes NLP models and facilitates interpretability, and AdapterHub Playground from Beck et al. (2022), a user-friendly platform for few-shot learning with language models. Nevertheless, we still observe limited options to collect human rationale and use it as feedback to improve a given model. This is especially true when it comes to tools that provide a visual UI for users without field expertise.

3.5.2 *Study* VIII †

In our study "IFAN: An Explainability-Focused Interaction Framework for Humans and NLP Models" (Mosca, Dementieva, et al., 2023), we investigate using explanations to enhance oversight on deployed models and align them more closely with human reasoning. Our work introduces a low-to-no-code framework that facilitates real-time explanation-based interaction with NLP models.

The *Interaction Framework for Artificial and Natural intelligence* (IFAN) web interface[1] is currently live and a quick video demo is available on YouTube[2]. Study VIII refers to the framework's state as of February 2023 (Mosca, Dementieva, et al., 2023) and its contribution can be summarized as follows:

(1) IFAN presents an interface that allows users without strong technical proficiency to contribute feedback to chosen NLP model explanations to rectify anomalies and unwanted behaviors. This feedback is subsequently incorporated through the use of efficient adapter layers.

(2) Additionally, our live platform provides a visual administration system and an API for managing models, datasets, and users, along with their respective access permissions.

(3) We showcase the effectiveness of our framework in reducing bias in a hate speech classifier and propose a feedback-rebalancing step to counteract the model's forgetfulness across multiple updates.

As illustrated in Figure 3.9, the platform comprises three main blocks (Mosca, Dementieva, et al., 2023, Section 3). The **Backbone** encompasses all machine learning

[1] https://ifan.ml/

[2] https://www.youtube.com/watch?v=BzzoQzTsrLo

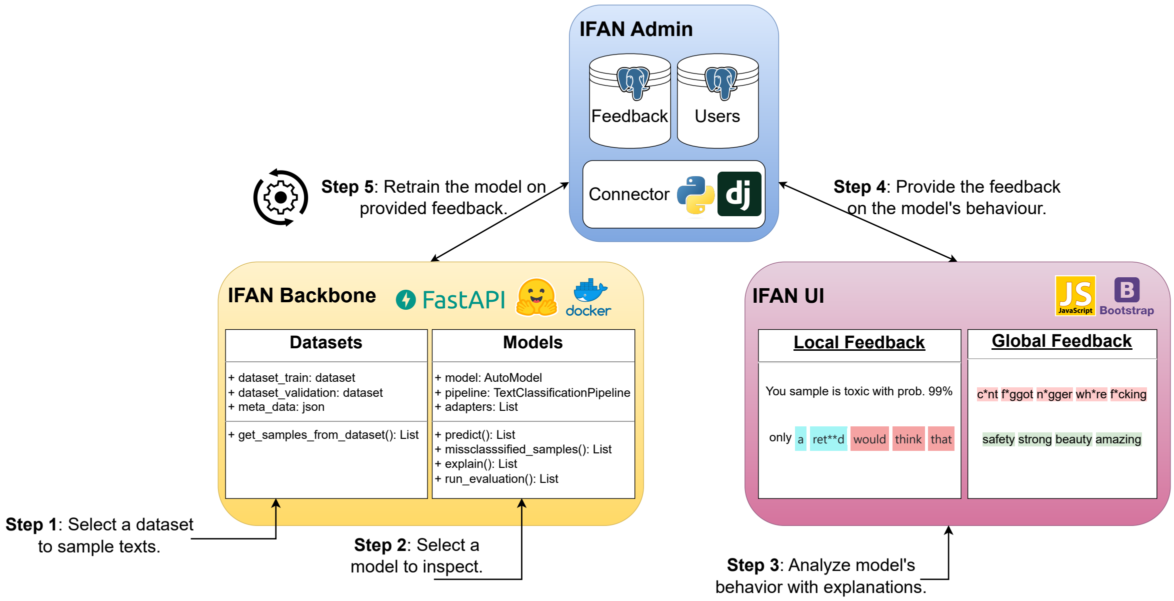

IFAN Admin
Feedback
Users
Connector
Step 5: Retrain the model on provided feedback.
Step 4: Provide the feedback on the model's behaviour.
IFAN Backbone
FastAPI
docker
Datasets
+ dataset_train: dataset
+ dataset_validation: dataset
+ meta_data: json
+ get_samples_from_dataset(): List
Models
+ model: AutoModel
+ pipeline: TextClassificationPipeline
+ adapters: List
+ predict(): List
+ missclasssified_samples(): List
+ explain(): List
+ run_evaluation(): List
IFAN UI
JS JavaScript
Bootstrap
Local Feedback
You sample is toxic with prob. 99%
only a ret**d would think that
Global Feedback
c*nt f*ggot n*gger wh*re f*cking
safety strong beauty amazing
Step 1: Select a dataset to sample texts.
Step 2: Select a model to inspect.
Step 3: Analyze model's behavior with explanations.

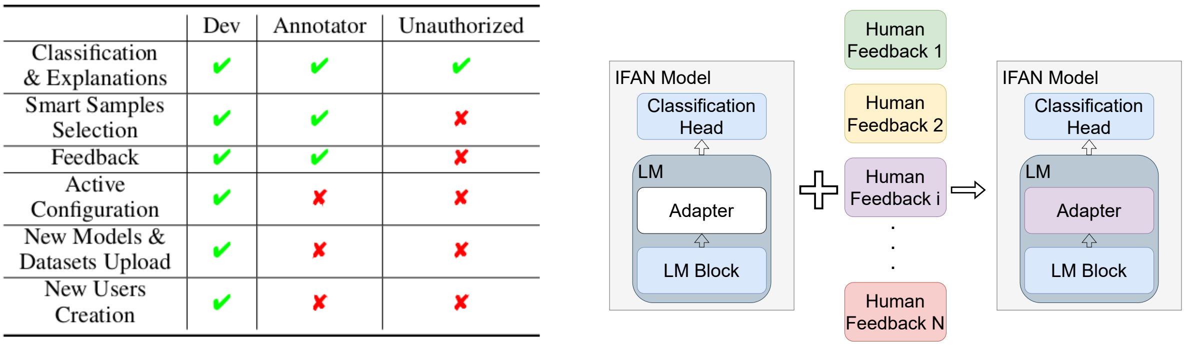
Figure 3.9: Overall structure of IFAN (Mosca, Dementieva, et al., 2023, Figure 2): (i) Users select a dataset or write a personalized input and (ii) select a model to be inspected. (iii) Through the user interface, annotators have the ability to review the model's prediction along with two kinds of explanations – local and global. (iv) Whenever the model exhibits irregular behavior, annotators have the opportunity to offer feedback. (v) The collected feedback is stored and subsequently utilized to fine-tune the model according to the human edits.

development elements, including datasets and models. We adhere to HuggingFace standard formats (Wolf et al., 2020) and encapsulate the entire backbone within a Docker[3] image for fast deployment.

The **User Interface** serves as the visual component of the platform, facilitating all human-machine interactions. It comprises of four main pages: *landing page* (home), *documentation*, *feedback*, and *configuration*. While the first three are available to all users, at least with limited functionalities, the last one is only accessible by authenticated developers. Here, developers can make use of additional visual resources to create and manage settings about models, datasets, users, and access rights.

Lastly, the **Admin** component manages the connection between the backbone and the user interface. It stores all user data, rights, and feedback-receiving samples in a

[3] https://www.docker.com



	Dev	Annotator	Unauthorized
Classification & Explanations	✔	✔	✔
Smart Samples Selection	✔	✔	✘
Feedback	✔	✔	✘
Active Configuration	✔	✘	✘
New Models & Datasets Upload	✔	✘	✘
New Users Creation	✔	✘	✘

(a) User Types and Rights

(b) Feedback Mechanism

Figure 3.10: Hierarchical access tiers to platform functionalities (a) (Mosca, Dementieva, et al., 2023, Table 1), and architecture of the NLP models integrated into IFAN (b). Adapter layers are added to each language model block and are trainable with human feedback (Mosca, Dementieva, et al., 2023, Figure 3).

PostgreSQL[4] database instance. Communication is handled through Python Django[5], which integrates all aspects related to user authentication, API calls/responses, state logs, and backbone resource locations.

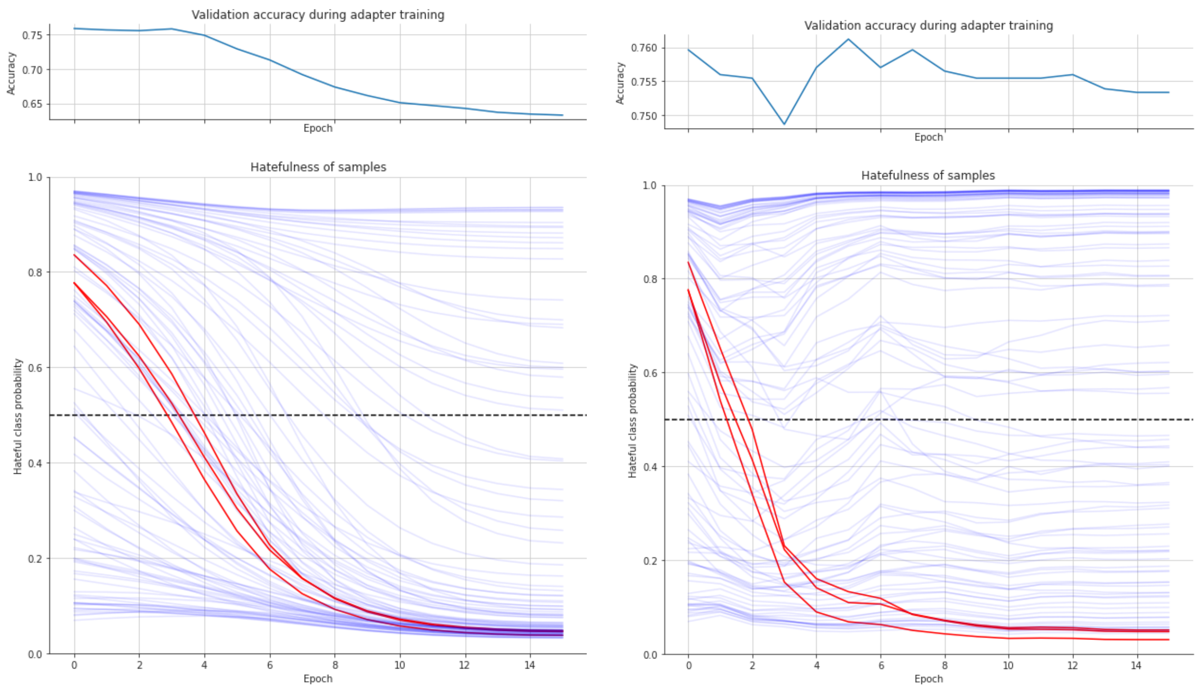
Users are categorized into three tiers: *developers*, *annotators*, and *unauthorized* users (see Figure 3.10a). Unauthorized users have limited access and are able to view model predictions and explanations, however their feedback is not considered. Annotators, with login credentials, can interact with the model, test it, view explanations, and provide feedback. Developers have full control over the platform, additionally managing users, roles, API access, models, and datasets.

At any time, IFAN specifies an *active dataset* and an *active model*. Feedback is incorporated into models using adapter layers (Houlsby et al., 2019), an emerging fine-tuning technique. Adapters are parameter-efficient layers added on top of each language model unit and are trained while freezing all other weights. If necessary, they can also be disabled to retrieve the original model state (see Figure 3.10b).

Users can evaluate the active model on the active dataset and correct any misclassifications. The platform provides local and global explanations, attributing scores using the LIME framework (M. T. Ribeiro, S. Singh, and Guestrin, 2016) and highlighting

[4] https://www.postgresql.org
[5] https://www.djangoproject.com

the most relevant tokens. Annotators can edit the highlighted tokens and send the updated explanation as feedback, which is then used to fine-tune the adapter layers.



(a) Fine-tuning without Rebalancing

(b) Fine-tuning with Rebalancing

Figure 3.11: Comparison between fine-tuning with and without feedback rebalancing: validation accuracy (top) and samples confidence (bottom). Samples receiving feedback are in red while the others are in blue. Rebalancing the feedback dataset provides a more stable run, without significantly impacting the overall learned knowledge of the model.

We apply the platform to a hate speech detection case study. Here, our aim is to debias a BERT model (Devlin et al., 2019) trained on the HateXplain dataset (Mathew et al., 2021), which showed bias towards samples targeting the Jewish subgroup. Results reveal that directly incorporating the feedback leads to model forgetfulness and thus to a substantial decrease in performance (Mosca, Dementieva, et al., 2023, Table 3). However, rebalancing the fine-tuning set by mixing the feedback with original samples mitigates the issue and allows us to effectively incorporate feedback while minimizing performance loss. Indeed, Figure 3.11 shows how the confidence of all samples goes down when the model is fine-tuned. Introducing rebalancing, instead, affects only samples receiving feedback (in red), while the performance on original texts (in blue) remains largely unchanged.

The full study can be found in Appendix B.4.

3.5.3 *Discussion*

The interpretability and controllability of modern NLP models are crucial for their ethical and safe use (European Commission, 2020). Study VIII contributes to these aspects by introducing IFAN—a real-time, explanation-based interaction framework designed for NLP models and human annotators. Its development is also driven by the need for tools that are more accessible by stakeholders without technical proficiency.

The framework incorporates feedback through adapter layers for efficient and iterative fine-tuning of models on specific tasks. Use case tests highlight IFAN's effectiveness in debiasing a hate speech classifier with minimal impact on performance.

As for the limitations, the feedback system is only designed for sequence-to-class applications. It is worth noting that the focus on classification settings is a common limitation among most XAI and EBHD approaches (Lertvittayakumjorn and Toni, 2021; Madsen, Reddy, and Chandar, 2022). At the same time, it still offers a limited range of explanations and feedback options. Lastly, our experiments have not yet identified clear patterns regarding the relationship between performance and feedback hyperparameters. Further research and testing are needed to determine the optimal number of feedback samples, fine-tuning epochs, and rebalancing ratio.

We remind the reader that study VIII refers to IFAN's state in February 2023, after which the team has continued working on addressing limitations as well as developing new features. We advocate for continuing research in this area, as it promotes broader and more diverse participation, enhancing fairness, transparency, and accountability.

4 DISCUSSION

The studies presented in this work contribute towards developing more interpretable, robust, and controllable models. This section provides an overarching discussion for each aspect together with key takeaways for the reader. At the same time, we reflect on our methodology's assumptions, successes, limitations, and learnings for future work.

4.1 INTERPRETABILITY

Improving the interpretability of NLP models is the first aspect addressed by this dissertation. Section 1.3 breaks down the broader goal into three specific objectives—i.e. (A) *assessing the applicability of explainability approaches to NLP*, (B) *tailoring explainers to NLP inputs*, and (C) *extending explanations to context-aware applications*.

Five studies address these objectives. Specifically, study I provides a thorough review of 41 SHAP-based explainability approaches. All methods are organized across five identified research directions and are examined under several criteria. Most importantly, for each method, the study contributes a concise yet comprehensive assessment regarding its suitability for NLP. Study II develops GrammarSHAP, a model-agnostic hierarchical explainer that can account for the sentence structure and dependencies between multi-word constituents. Finally, studies III, IV, and V work on applying explainability to NLP models that account for context in the form of additional input modes. The first two carry out experiments on hate speech detection for tweets coupled with user and network data, whereas the third works on survey data presenting both structured and unstructured answer formats.

Our methodology's successes are evident as it provides valuable guidelines for those involved in NLP research and practice (study I), demonstrates the adaptability of explainers for natural language inputs (study II), and extends current approaches to context-sensitive multi-modal applications (studies III, IV, and V). Nonetheless, it remains important to acknowledge its limitations.

First and foremost, the proposed methodology is predominantly tailored to sequence-to-class tasks. Such a focus reflects the existing literature and provides a useful simplification for research. However, it also reveals a gap between XAI and more complex sequence-to-sequence tasks. While many feature attribution methods can be applied iteratively and thus extended to more complex tasks, the necessary steps are often not straightforward and mark an underexplored area of this work and—more generally—XAI research.

Secondly, our studies heavily rely on the SHAP framework. This choice is motivated by SHAP's solid theoretical foundation, its general applicability, and its popularity across several domains. However, we recognize that this reliance on SHAP may limit the scope of our research. Future research could benefit from scrutinizing alternative frameworks and considering new emerging ideas as a foundation for their work.

Lastly, several of our reviewed or proposed approaches may lead to high computational efforts, particularly when dealing with long texts. This is due to their iterative nature, which involves processing one token (or a few) at a time. In practice, this may result in slower running times despite scaling linearly with the input length.

Takeaways:

- Model interpretability is necessary and leads to a much deeper understanding than merely inspecting performance metrics.
- Based on the existing literature, looking at XAI approaches beyond the NLP field is a must (and very fruitful) to keep progressing in the field.
- Post-explainability frameworks can be adapted and extended to fit NLP applications and relevant input characteristics thereof—text structure, word interactions, multi-modality, and context-dependency.

- XAI can turn models into a source of knowledge, but we are far from one-fits-all interpretability approaches. Combining complementary explanation approaches and formats is a great alternative so far.
- There is a strong focus on sequence-to-class applications, making the extension to sequence-to-sequence the priority for future work.

4.2 ROBUSTNESS

Section 1.3 narrows down the broad scope of improving robustness to the objective of (D) *detecting adversarial attacks via model explanations*. We argue that adversarial detection is more profitable than standard defenses as attacking attempts can be recognized explicitly. Such ability enables developers to collect valuable manipulated samples as well as identify adversarial third parties. This dissertation presents two studies contributing to this objective.

Both study VI and study VII investigate the usage of instance-level model explanations to detect word-level adversarial text attacks. Both works train detectors on a large number of original and manipulated samples alongside their explanation signatures. Such detectors learn to recognize patterns signaling adversarial perturbations, therefore creating an additional layer of defense when deployed together with the targeted model. The first work extracts such signals through SHAP explanations whereas the second proposes an ad-hoc custom metric (WDR) to quantify the model's reaction to the input.

Our works further demonstrate the strong link between model interpretability and robustness discussed in section 2.6. Furthermore, especially with the improvements introduced by study VII, our proposed methodology substantially outperforms existing methods and contributes to progress in the field. One of the main successes is the strong generalization capabilities that XAI-based adversarial detectors show. At the same time, they are model- and dataset-agnostic while allowing for a high degree of customization.

As partially discussed already in the methodology, among the limitations we acknowledge the primary focus on word-level attacks or rather on attacks relying on a few token replacements. Also, our approaches' running time scales linearly with the input length, which may be undesirable when dealing with particularly long texts. Finally, while our studies stand out in performance, future work could benefit from a more thorough comparison using a wider array of metrics specifically designed for this purpose.

Viewed from a broader standpoint, defense strategies can potentially inspire and stimulate novel and improved attack techniques. A case in point is BAE (Garg and Ramakrishnan, 2020), capitalizing on more resilient architectures like BERT to generate highly-effective contextual attacks. In the context of our work, the proposed defense strategies could lead to novel adaptive attacks operating at the sentence level.

Takeaways:

- Interpretability and robustness are deeply connected. Anomalous behavior in one aspect has direct implications on the other.
- Model explanations carry strong and easy-to-read patterns to explicitly detect adversarial examples. Following this intuition, model/task-agnostic detectors can be trained and deployed alongside models.
- Looking beyond feature attribution explanations is necessary for future research to avoid overspecializing on word-level attacks and overlooking sentence-level manipulations.

4.3 HUMAN OVERSIGHT

In the context of this work, we narrow down the aspect of controllability to the objective of (E) *enabling model controllability through human explanations* (see section 1.3). More in detail, we work on human-model interaction pipelines enabling annotators to influence and control deployed models via feedback. Special consideration is given to stakeholders with no technical proficiency and thus to the user-friendliness of the framework's design.

We contribute one study (and a prototype) to this objective. Study VIII proposes IFAN—a real-time explanation-based framework for the interaction between models and human annotators. Through its UI, users can evaluate an active model, inspect predictions and explanations, and edit them to provide feedback and steer the model's behavior accordingly. Beyond that, IFAN's live platform provides a visual admin system and API to effectively manage models, datasets, as well as users and their access rights.

Overall, results show the methodology to be effective at iteratively incorporating feedback into models through adapter layers. Experiments in debiasing a hate speech classifier show that mixing feedback samples with original ones mitigates the issue of model forgetfulness and drastically reduces the impact on performance. However, while preserving performance as we update and control the model is a success, human feedback is rarely capable of further improving predictive capabilities in terms of standard metrics.

From a broader perspective, there are also other challenges and aspects that should be considered for future development of frameworks like IFAN. Preserving high-quality of human feedback is complex and vulnerable to misuse by adversarial agents, especially when a small group of annotators can influence the model significantly (Al Kuwatly, Wich, and Groh, 2020). Future work should incorporate a strict management system like IFAN's and track annotators' impact to increase trustworthiness.

Poor models and interfaces can also lead to user frustration and affect feedback quality (Lertvittayakumjorn and Toni, 2021). The issues can be mitigated by using state-of-the-art models and through user studies to design suitable interfaces. On the other hand, convincing explanations may lead to an overestimation of a model's capabilities, causing misplaced trust. To address this, we suggest providing a diverse range of explanations for users (Madsen, Reddy, and Chandar, 2022) and detailed model reports for developers, offering a comprehensive understanding of the models to be deployed.

Takeaways:

- Explanations are a great channel for models and humans to interact.
- Incorporating feedback with parameter-efficient tuning can positively influence a model to be in line with our rationale, but is unlikely to improve the overall performance by much.
- Model controllability is a challenge that needs to be addressed holistically and requires efforts from many perspectives: meaningful explanations, effective feedback incorporation, user-friendly interface design, strict security features, and ad-hoc resources management tools.
- When developing an interaction framework, accurate models and well-designed interfaces are not negotiable—even at the inception stage.

5 CONCLUSION

This dissertation is a step towards a more human-centric development of NLP models, emphasizing the key requirements of (1) interpretability, (2) robustness, and (3) human oversight. We place explainable artificial intelligence at the core of our methodology, leveraging model explanations to pursue all three goals.

A total of eight studies were presented in this work. Five studies (I-V) contribute to the aspect of model interpretability across three objectives. The first study reviews SHAP-based methods with the aim of (A) *assessing the applicability of explainability approaches to NLP*. The second study proposes GrammarSHAP as an instance of (B) *tailoring explainers to NLP inputs*. The remaining three studies work on (C) *extending explainability to context-aware applications* by interpreting multi-modal models leveraging context in conjunction with text.

Two studies (VI-VII) contribute to improving model robustness. This is achieved by developing state-of-the-art model-agnostic approaches for (D) *detecting adversarial attacks via model explanations*. Finally, one study (VIII) contributes to improving human oversight. To this end, it develops IFAN, an interaction framework between NLP models and human annotators with the aim of (E) *enabling model controllability through human explanations*.

Our work shows the potential in terms of post-hoc explainability approaches available and how they can add value to NLP research. At the same time, methods can be adapted and extended meaningfully for NLP applications, even when these entail context through multi-modal inputs. Our methodology also reveals that model explanations connect interpretability, robustness, and controllability as interacting dimensions of human-centricity. We demonstrate that anomalies in model explanations are directly correlated with adversarial manipulations. Based on this principle, our work develops state-of-the-art model-agnostic adversarial detectors, providing future research with a strong baseline to defend models against word-level attacks. Finally, results collected from IFAN show that editing explanations can be utilized to incorporate feedback into NLP models—fixing undesired outputs and effectively controlling their behavior according to human intentions.

We strongly encourage future research to continue work on the key requirements to achieve human-centric NLP systems. Our findings suggest to not explore the various aspects in isolation, but rather keep investigating their interconnections while incorporating additional ones—e.g. *fairness*, *privacy*, and *accountability*. Following a human-centric approach should be at the heart of future AI research. By focusing on the needs and values of human users, we can develop systems that not only perform well, but are also trusted and accepted by the people they serve.

APPENDIX

A PUBLICATIONS •

Publications in Appendix A are relevant for examination in accordance with Exhibit 6 of the regulations for the award of the doctoral degree.

A.1 STUDY I

Edoardo Mosca, Ferenc Szigeti, Stella Tragianni, Daniel Gallagher, and Georg Groh (Oct. 2022). "SHAP-Based Explanation Methods: A Review for NLP Interpretability." In: *Proceedings of the 29th International Conference on Computational Linguistics*. Gyeongju, Republic of Korea: International Committee on Computational Linguistics, pp. 4593–4603. URL: https://aclanthology.org/2022.coling-1.406

[1] https://creativecommons.org/licenses/by/4.0/

Publication Summary

"Model explanations are crucial for the transparent, safe, and trustworthy deployment of machine learning models. The SHapley Additive exPlanations (SHAP) framework is considered by many to be a gold standard for local explanations thanks to its solid theoretical background and general applicability. In the years following its publication, several variants appeared in the literature—presenting adaptations in the core assumptions and target applications. In this work, we review all relevant SHAP-based interpretability approaches available to date and provide instructive examples as well as recommendations regarding their applicability to NLP use cases." (Mosca, Szigeti, et al., 2022, p. 1)

Author Contributions

Edoardo Mosca contributed to the study as follows:

- Conception, development, and lead of the research project **100%**
- Review work **60%**
- Structure and assessment of existing literature **50%**
- Drafting of the manuscript **60%**
- Submission, peer review, and publication Process **90%**

SHAP-Based Explanation Methods: A Review for NLP Interpretability

Edoardo Mosca
TU Munich,
Department of Informatics,
Germany
edoardo.mosca@tum.de

Ferenc Szigeti
TU Munich,
Department of Informatics,
Germany
ferenc.szigeti@tum.de

Stella Tragianni
TU Munich,
Department of Informatics,
Germany
stella.tragianni@tum.de

Daniel Gallagher
University College Dublin,
Department of Informatics,
Ireland
daniel.gallagher1@ucdconnect.ie

Georg Groh
TU Munich,
Department of Informatics,
Germany
grohg@in.tum.de

Abstract

Model explanations are crucial for the transparent, safe, and trustworthy deployment of machine learning models. The *SHapley Additive exPlanations* (SHAP) framework is considered by many to be a gold standard for local explanations thanks to its solid theoretical background and general applicability. In the years following its publication, several variants appeared in the literature—presenting adaptations in the core assumptions and target applications. In this work, we review all relevant SHAP-based interpretability approaches available to date and provide instructive examples as well as recommendations regarding their applicability to NLP use cases.

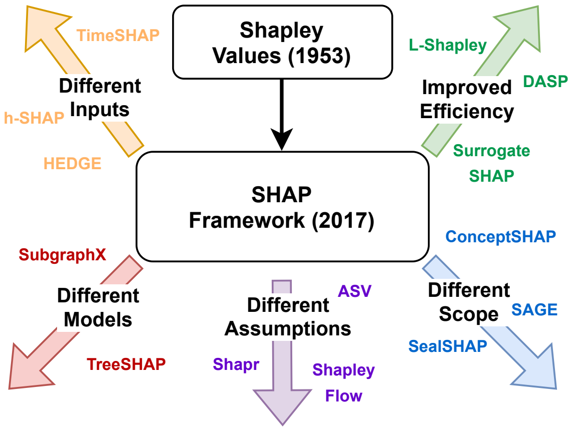


Figure 1: This work identifies five research directions pursued by Shapley- and SHAP-based approaches in XAI. Each direction, together with a few notable methods as examples, has been indicated by a different color.

1 Introduction

Several methods have been proposed to address the issue of opacity in modern machine learning models. Most notoriously, explanations are fundamental for *Deep Neural Networks* (DNNs) (Devlin et al., 2019; Madsen et al., 2021; Mosca et al., 2021) as these automatically learn millions of parameters and behave like black-boxes. Lundberg and Lee (2017) proposes *SHapley Additive exPlanations* (SHAP), a unified local-interpretability framework with a rigorous theoretical foundation on the game-theoretic concept of Shapley values (Shapley, 1953).

SHAP is nowadays considered a core contribution to the field of *eXplainable Artificial Intelligence* (XAI). Following its publication, a variety of explainability approaches based on SHAP's methodology has populated the literature and this trend continues to grow. Some present a new version of SHAP tailored to a certain type of input data—e.g. graphs (Yuan et al., 2021) and text (Chen et al., 2020)—or to specific models such as random forests (Lundberg et al., 2018). Others, instead, modify SHAP's underlying assumptions—e.g. features independence—to increase the original framework's flexibility for cases in which they are too strict or overly simplistic (Frye et al., 2019).

In this work, we **(1)** identify five broad research directions inspired by SHAP, **(2)** review available SHAP-based (or Shapley-value-based) approaches as members of such categories, and **(3)** investigate their applicability in the domain of *Natural Language Processing* (NLP).

Our work reviews 41 methods with a particular focus on their core assumptions, input require-

Proceedings of the 29th International Conference on Computational Linguistics, pages 4593–4603
October 12–17, 2022.

ments, explanation form, and available implementations. Furthermore, we provide NLP researchers with use-case-based recommendations and instructive examples.

2 Background

For the sake of clarity, we provide a gentle introduction to Shapley values and the methods for their estimation, most notably SHAP. All concepts will be explained informally, resorting to formalities when necessary.

2.1 Shapley Values

Shapley Values are a concept from game theory, originally developed as a measure to fairly distribute a reward among a set of players contributing to a certain outcome (Shapley, 1953). In the context of machine learning models, the players involved are the input features and the outcome is the model's decision, Shapley values attribute an importance score to each part of the input (Lundberg and Lee, 2017).

Given the set of input features $\mathbf{F} = \{1, 2, \ldots, p\}$, all features in a certain coalition $S \subseteq \mathbf{F}$ cooperate towards the outcome $val(S)$—with the default $val(\emptyset) = 0$. Shapley values redistribute the total outcome value $val(\mathbf{F})$ among all features based on their average marginal contribution across all possible coalitions S. More specifically, feature i's marginal contribution w.r.t. a coalition S:

$$\Delta_{val}(i, S) = val(S \cup \{i\}) - val(S)$$

is averaged across all $S \subseteq \mathbf{F} \setminus \{i\}$. Hence, the corresponding Shapley values $\phi_{val}(i)$ measures its contribution based on the formula:

$$\phi_{val}(i) = \sum_{S \subseteq \mathbf{F} \setminus \{i\}} \frac{|S|!(p - |S| - 1|)!}{p!} \Delta_{val}(i, S)$$

Here, the coefficient $\frac{|S|!(p-|S|-1|)!}{p!}$ is used as normalization term based on the number of choices for the subset S. This redistribution of the total outcome $val(\mathbf{F})$ respects the four properties of:

Efficiency: All features contributions add up to the total outcome, i.e. $\sum_{i \in \mathbf{F}} \phi_{val}(i) = val(\mathbf{F})$.

Symmetry: If $val(S \cup \{i\}) = val(S \cup \{j\})$ for all $S \subseteq \mathbf{F} \setminus \{i, j\}$, then $\phi_{val}(i) = \phi_{val}(j)$

Dummy: If $val(S \cup \{i\}) = val(S)$ for all $S \subseteq \mathbf{F}$, then $\phi_{val}(i) = 0$

Additivity: In the presence of a single game with two outcomes val_1 and val_2, then Shapley values are additive w.r.t. the combined outcome, i.e. $\phi_{val_1+val_2}(i) = \phi_{val_1}(i) + \phi_{val_2}(i)$

2.2 Shapley Values Approximation and SHAP

The idea of utilizing Shapley values to compute feature attribution scores precedes the SHAP framework (Lipovetsky and Conklin, 2001; Song et al., 2016). In this case, the outcome val of the game is the prediction of a machine learning model f and Shapley values $\phi_f(i)$ measure the influence that each feature i has based on its current value. The early literature also worked on approximation strategies, as the exponential number of coalitions renders the exact estimation of Shapley values unfeasible (Štrumbelj and Kononenko, 2014; Datta et al., 2016). The main idea from these works is to compute $\phi_f(i)$ only for a smaller selection of subsets $S \subseteq \mathbf{F}$ and to estimate the effect of removing a feature by integrating over training samples. This eliminates the need to retrain the model for each choice of S.

The work from Lundberg and Lee (2017) introduces a new perspective that unifies Shapley value estimation with popular explainability methods such as LIME (Ribeiro et al., 2016), LRP (Binder et al., 2016), and DeepLIFT (Shrikumar et al., 2017). Furthermore, they propose SHAP values as a unified measure of feature importance and prove them to be the unique solution respecting the criteria of *local accuracy*, *missingness*, and *consistency*. The authors contribute a library of methods to efficiently approximate SHAP values in a variety of settings:

KernelSHAP: Adaptation of LIME—hence model-agnostic—to approximate SHAP values. As it works for any model f, it cannot make any assumption on its structure and is thus the slowest within the framework.

LinearSHAP: Specific to linear models, uses the model's weight coefficients and optionally accounts for inter-feature correlations.

DeepSHAP: Adaptation of DeepLIFT—hence specific to neural networks–to approximate SHAP values. Considerably faster than its model-agnostic counterpart as it makes assumptions about the model's compositional nature.

While not initially presented in Lundberg and Lee (2017), the following algorithms were later

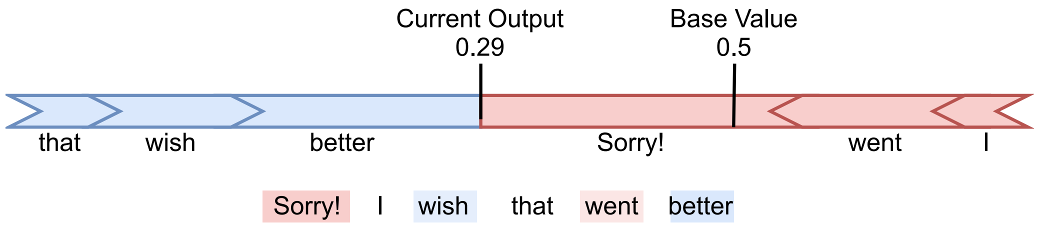


Figure 2: Example of explanation for sentiment analysis that can be generated with the SHAP library, e.g. with KernelSHAP. The base value indicates the model's average prediction. Each feature—i.e. word—contributes to the outcome, thus justifying the difference between the average and the current outcome.

added as part of the framework:

PartitionSHAP: Faster version of KernelSHAP that hierarchically clusters features. This hierarchy defines feature coalitions based on their interactions.

GradientSHAP: An extension of the *Integrated Gradients* (IG) method (Sundararajan et al., 2017)—again specific to neural networks—that aggregates gradients over the difference between the expected model output and the current output.

TreeSHAP: A fast method for computing exact SHAP values for both trees and ensembles (Lundberg et al., 2020a). In comparison to KernelSHAP, it also accounts for interactions among features.

Other minor approaches—PermutationSHAP, SamplingSHAP, ExactSHAP, and MimicSHAP—are also available in the official library[1]. To avoid confusion, we point out that the implementations have slightly different names: they use *"Explainer"* instead of *"SHAP"*. For instance, KernelSHAP and DeepSHAP are implemented with the names of *KernelExplainer* and *DeepExplainer* respectively. Figure 2 sketches an explanation generated with SHAP.

3 Search and Selection Criteria

As the popularity of SHAP increases, also the number of approaches based on it or directly on Shapley values has been on the rise. In fact, $\sim 3,200$ of the $\sim 6,900$ papers citing Lundberg and Lee (2017) are from 2021, an exponential increase when compared to previous years (1563, 567, and 118)[2].

Besides the papers already known to us, we manually screened all works citing SHAP with at least 15 citations[2]. This systematical search, based on the assumption that SHAP-based approaches should at least reference Lundberg and Lee (2017), helped us uncover several relevant contributions and mitigate the selection bias induced by our previous knowledge. The threshold of 15 citations was introduced to speed up our manual search and to filter out works that have not received the research community's attention. To account for temporal bias—i.e. that publications accumulate citations over time—we lowered the threshold to 10 for papers published in the most recent years (2021 and 2022)[2]. We only consider and review papers that contributed new SHAP-based approaches and exclude those—like (Wang, 2019) and (Antwarg et al., 2019)—utilizing SHAP (almost) off-the-shelf. Similarly, we exclude works such as Wang et al. (2020) and Huber et al. (2022) utilizing Shapley values for purposes not directly connected with explainability.

4 Existing Reviews

Previous reviews like Linardatos et al. (2021), Vilone and Longo (2020), and Madsen et al. (2021) present extensive overviews of explainability methods, but only briefly mention SHAP and a few of its derivates. Others—such as Covert et al. (2021), Sundararajan and Najmi (2020), and Kumar et al. (2020)—review some Shapley-based methods in detail (between 5 and 9) but do not construct a comprehensive review. Our work, in contrast, significantly extends this range and covers more than 40 approaches.

5 Review: SHAP-Based Approaches

Several works proposed methods based on SHAP, or more generally on Shapley values, following the contribution from Lundberg and Lee (2017). While the changes and variations introduced have been at times criticized for not being as rigorous as SHAP in following its core assumptions (Sundararajan

[1] https://github.com/slundberg/shap

[2] All queries are performed with Google Scholar. Accessed on 10.05.2022.

and Najmi, 2020), SHAP-based methods continue to increase in both quantity and popularity.

Our review categorizes SHAP-based approaches available to date based on *how they differ from* and *how they improve on* the original SHAP framework. We identify five broad categories in the existing literature, each one of them describing a different research direction pursued by its members:

(C1) Tailored to Different Input Data: This category contains approaches specialized on specific input data structures such as graphs (Wang et al., 2021), structured text (Chen et al., 2020), and images (Teneggi et al., 2021). In some cases, approaches are used complementary for applications dealing with multimodal inputs (Wich et al., 2021; Mosca et al., 2022b).

(C2) Explaining Different Models: Methods in this class are specifically designed to explain predictions from particular types of machine learning models such as random forests (Lundberg et al., 2018; Labreuche and Fossier, 2018) and neural networks (Ghorbani and Zou, 2021). Hence, these are model-specific.

(C3) Modifying Core Assumptions: SHAP treats features as independent. Newer methods offer the possibility to account for dependencies between features (Frye et al., 2019) and for causal structures behind their interactions (Heskes et al., 2020).

(C4) Producing Different Explanations Types: SHAP is a framework for local feature-attribution explanations, i.e. it attributes scores to input components based on their instance-level contributions. Methods in this category have a different scope and generate explanations that convey a different type of information. This can vary from global explanations (Covert et al., 2020) to counterfactual explanations (Singal et al., 2019) and concept explanations (Yeh et al., 2020).

(C5) Estimating Shapley Values More Efficiently: These approaches comprise alternative strategies for the approximation of Shapley values. Their focus is on leveraging prior knowledge about the data and model to improve the approximation *efficiency* and *accuracy* (Messalas et al., 2019; Chen et al., 2018).

Clearly, these categories are not designed to be exclusive. Therefore, an approach can fall in more than one if it differs from SHAP in multiple aspects. Table 1 provides an overview of all approaches with their main characteristics. As one can observe, the majority of approaches are identified as part of more categories, i.e. research directions.

5.1 Approaches Tailored to Different Inputs

SHAP does not make strong assumptions on the target model's input. While this suggests that it is suitable for all input types, its lack of specificity results in limitations when applied directly to different inputs than tabular data.

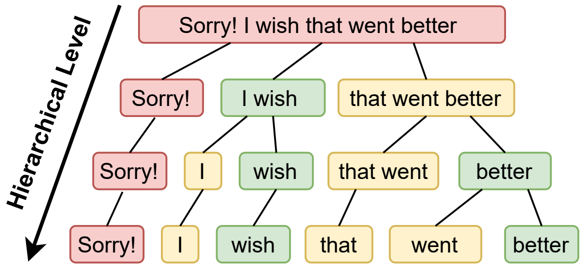


Figure 3: Example of hierarchical explanation that can be generated with HEDGE (Chen et al., 2020) for a sentiment analysis model. Each token is colored by contribution: negative (red), neutral (yellow), and positive (green). Going one level lower represents a token-breakdown step and thus more fine-grained Shapley values.

For text data, only measuring each individual feature's effect is an oversimplification, as words present strong interactions and their meaning and contribution heavily rely on the context. Thus, when it comes to text data, only considering single words as features is quite restrictive and relevance scores should be applied to multi-level tokens or even to entire sentences. *Hierarchical Explanation via Divisive GEneration* (HEDGE) (Chen et al., 2020) is an example of a SHAP-based method addressing this issue for (long) texts. Based on the weakest token interactions, it iteratively divides the text into shorter phrases and words in a top-down fashion. At each level, a relevance score is attributed to each token, resulting in a hierarchical explanation (Chen et al., 2020). PartitionSHAP, recently added to the official SHAP repository[3], follows a similar strategy by creating hierarchical features coalitions and measuring their interactions.

[3] https://github.com/slundberg/shap

Method	**Categories**	**Description**	**NLP Applicability / Implementation**
SHAP (Lundberg and Lee, 2017)		The original SHAP framework including the methods: KernelSHAP, LinearSHAP, DeepSHAP, etc.	*Ready Off-the-Shelf* Python
AVA (Bhatt et al., 2020)	**(C5)**	Combines the explanations of nearest neighbors to explain a given instance	*Adaptable* n.a.
ASV (Frye et al., 2019)	**(C1) (C3)**	Relaxes the symmetry axiom of Shapley values to incorporate causal structure into explanations	*Potentially Applicable* R
BShap (Sundararajan and Najmi, 2020)	**(C4) (C5)**	Baseline approach to facilitate comparison between different Shapley value based methods	*Adaptable* n.a.
C- and L-Shapley (Chen et al., 2018)	**(C3) (C5)**	Efficient feature attribution method that models data as a graph by considering only neighboring features	*Ready Off-the-Shelf* TensorFlow
CASV (Singal et al., 2019)	**(C1) (C2) (C3) (C4)**	Shapley value adaptation to account for counterfactuals by adhering to the Rubin Causal Model	*Not Relevant* n.a.
Causal Shapley (Heskes et al., 2020)	**(C1) (C3)**	Computing feature importance on data with (partial) causal ordering using Pearl's do-calculus	*Potentially Applicable* R
ConceptSHAP (Yeh et al., 2020)	**(C4)**	Unsupervised discover of concepts inherent to the data and model based on Shapley values	*Ready Off-the-Shelf* PyTorch
DASP (Ancona et al., 2019)	**(C3) (C5)**	Polynomial-time approximation of Shapley values in DNNs	*Adaptable* TensorFlow
Data Shapley (Ghorbani and Zou, 2019)	**(C4)**	Shapley-based importance attribution method for individual data instances in the training set	*Potentially Applicable* TensorFlow
DeepSHAP v2 (Chen et al., 2021)	**(C2) (C5)**	Computes efficiently SHAP values for DNNs with an extension to explain stacks of mixed model types	*Adaptable* n.a.
GrammarSHAP (Mosca et al., 2022a)	**(C1) (C3)**	Hierarchical explanations for text inputs based on the sentence grammatical structure	*Adaptable* n.a.
gSHAP (Tan et al., 2018)	**(C4)**	Generates intuitive Shapley-based global by aggregating local explanations	*Potentially Applicable* n.a.
h-SHAP (Teneggi et al., 2021)	**(C1) (C5)**	Hierarchical implementation of Shapley values for their efficient computation in image data	*Potentially Applicable* PyTorch
HEDGE (Chen et al., 2020)	**(C1) (C3)**	Hierarchical explanations based on feature interaction detection specifically for text data	*Ready Off-the-Shelf* PyTorch
Integrated Hessians (Janizek et al., 2021)	**(C5)**	Extension of Integrated Gradients to explain pairwise feature interactions in NNs	*Ready Off-the-Shelf* PyTorch
lossSHAP (Lundberg et al., 2020b)	**(C2) (C4)**	Obtain global explanations by aggregating local explanations with TreeSHAP	*Potentially Applicable* Python
MCDA Explainer (Labreuche and Fossier, 2018)	**(C1) (C2) (C3)**	Proposes the *influence index*, which is an extension of Shapley values for MCDA tree models	*Not Relevant* n.a.
Neuron Shapley (Ghorbani and Zou, 2021)	**(C2) (C4)**	Quantifies the contributions of single neurons to single predictions and overall model performance	*Adaptable* TensorFlow
R2 decomposition (Redell, 2019)	**(C5)**	Feature importance attribution based on Shapley value variance decomposition	*Potentially Applicable* R
Shapley Flow (Wang et al., 2021)	**(C1) (C3)**	Enables the addition of a causal graph encoding relationships among input features	*Potentially Applicable* Python
SAGE (Covert et al., 2020)	**(C4) (C5)**	Efficiently quantifies each feature's contribution to the model's performance for global explainability	*Potentially Applicable* Python
SealSHAP (Parvez and Chang, 2021)	**(C4)**	Shapley-based usefulness measure of individual data sources for transfer learning	*Ready Off-the-Shelf* TensorFlow
Shap-C (Ramon et al., 2019)	**(C4) (C5)**	Combination of computing counterfactuals and Shapley Values	*Potentially Applicable* Python
Shapley Residuals (Kumar et al., 2021)	**(C4)**	Captures information lost by KernelSHAP in Shapley Residuals, which characterize feature dependence	*Potentially Applicable* n.a.
Shapley Taylor index (Dhamdhere et al., 2020)	**(C3) (C5)**	Generalization of the Shapley value that attributes the model's prediction to interactions of subsets of features	*Potentially Applicable* n.a.
Shapr (Aas et al., 2021)	**(C3)**	Extends KernelSHAP to handle data with dependent features and produce more realistic explanations	*Potentially Applicable* R
SPVIM (Williamson and Feng, 2020)	**(C4) (C5)**	Global variable importance measure using an efficient regression-based Shapley value estimator	*Not Relevant* Python and R
SubgraphX (Yuan et al., 2021)	**(C1) (C2) (C5)**	Explain GNNs by identifying important subgraphs using Shapley values as importance measures	*Not Relevant* PyTorch
SurrogateSHAP (Messalas et al., 2019)	**(C5)**	An XGBoost tree model is trained as a surrogate model on the target model and TreeSHAP is applied to explain it	*Potentially Applicable* n.a.
TreeSHAP (Lundberg et al., 2018)	**(C2) (C5)**	Fast and exact method to estimate SHAP values for tree models and ensembles of trees	*Potentially Applicable* Python
TimeSHAP (Bento et al., 2021)	**(C1) (C2) (C4)**	Adapts KernelSHAP to sequential data and produces feature, event and cell-wise explanations	*Potentially Applicable* n.a.

Table 1: Overview of available Shapley- and SHAP-based methods. For each method we also indicate the categories it belongs to, its main idea and intuition, and its applicability to NLP together with the available implementations. See 6.1 for more details about our NLP-applicability assessment.

Figure 3 sketches an example of a hierarchical explanation for text data.

For models trained on graph data, especially graph DNNs, Yuan et al. (2021) proposed to explain predictions by using Shapley values as a measure of subgraph importance. The resulting method—named SubgraphX—also captures the interactions between different subgraphs.

On images, SHAP can face computational limitations as the number of features, i.e. pixels, can become extremely large. h-SHAP (Teneggi et al., 2021) efficiently retrieves exact Shapley values by hierarchically excluding irrelevant image areas from the computation. This is done following the observation that, if a certain area in the image is uninformative, so are its constituent sub-areas, which are therefore not worth exploring.

5.2 Approaches Explaining Different Models

Explanation methods making fewer assumptions on the target classifier benefit from better applicability as they can explain a wider range of models. However, this can hinder explanations in terms of accuracy, information granularity, and computational efficiency. As we have already seen in 2.2: KernelSHAP has the key advantage of being model-agnostic, but it is drastically more inefficient than its DNN-specific counterpart DeepSHAP (Lundberg and Lee, 2017).

An example of a highly-specialized explainability method is TreeSHAP, presented by Lundberg et al. (2018) as an extension of the SHAP framework. This approach, only applicable to decision trees or ensembles thereof, is a highly efficient algorithm for exact SHAP values retrieval. Not only the approach needs considerably less computational effort than the more general variants such as KernelSHAP, but it leverages the decision tree structure to compute SHAP interaction values and thus captures pairwise interactions between features.

Ghorbani and Zou (2021) proposes *Neuron Shapley*, a framework targeting DNN models which is able to quantify each individual neuron's contribution to single predictions and overall model performance. An example of the kind of explanation enabled by Neuron Shapley is visualized in figure 4. By analyzing interactions between neurons and picking those which exhibit the largest Shapley value, this method is particularly suitable for identifying neurons responsible for biases and vulnerabilities (Ghorbani and Zou, 2021).

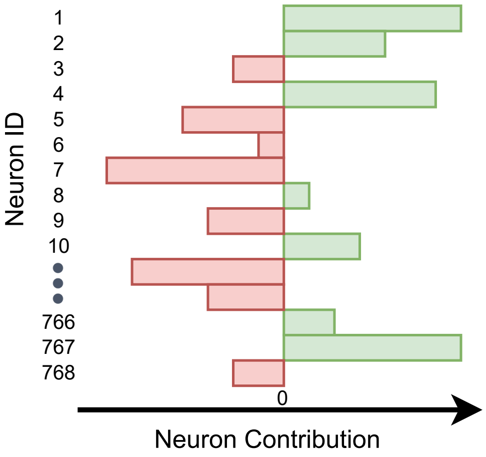


Figure 4: Sketch of a Neuron Shapley explanation for the 768 neurons of BERT output layer (Devlin et al., 2019). A Shapley value is assigned to each neuron depending depending on how they contribute towards the prediction (green) or against it (red).

5.3 Approaches Modifying Core Assumptions

Assumptions made by SHAP can be at times too restrictive or simplistic, which can prevent explanations from accessing and leveraging crucial information such as dependency relationships between input features. For instance, already the symmetry property of Shapley values treats features as independent. While this can be true in some cases, for instance when dealing with tabular data with uncorrelated variables, it is an oversimplification when it comes to texts, images, and more structured data.

Frye et al. (2019) introduces *Asymmetric Shapley Values* (ASV), which drops the symmetry assumption and enables the generation of model-agnostic explanations incorporating any causal dependency known to be present in the data. Similar approaches are:

- *Causal Shapley* (Heskes et al., 2020), additionally requiring a partial causal ordering of the features as input.
- *Shapley Flow* (Wang et al., 2021), which leverages a causal graph, encoding relationships among input features.
- *Shapr* (Aas et al., 2021), an extension of KernelSHAP relaxing the feature independence assumption.

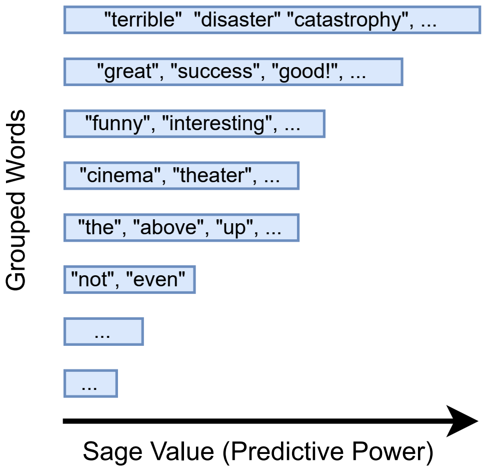

"terrible" "disaster" "catastrophy", ...
"great", "success", "good!", ...
"funny", "interesting", ...
"cinema", "theater", ...
"the", "above", "up", ...
"not", "even"
...
...
Grouped Words
Sage Value (Predictive Power)

Figure 5: Example of SAGE explanation for a sentiment analysis model. Since the number of global features is as large as the vocabulary, words need to be grouped together (e.g. by similarity) to reduce the number of features to be explained.

5.4 Approaches Producing Different Explanation Types

The SHAP framework and many of its derivatives mainly focus on generating local explanations based on feature importance. However, the general applicability of Shapley values combined with its strong foundations also offers potential for different explainability settings. More recent works have explored the usage of Shapley values to build other types of explanations conveying different kinds of information about the model and the available data.

For instance, *Data Shapley* (Ghorbani and Zou, 2019) estimates the importance of each training sample for a given machine learning model. Similarly, SealSHAP (Parvez and Chang, 2021) attributes usefulness scores to data sources for transfer learning.

Covert et al. (2020) introduces *Shapley Additive Global importancE* (SAGE), an explainability method analogous to SHAP but with a core focus on global explainability. More in detail, SAGE is a model-agnostic method that quantifies the predictive power of each input feature for a given model while also accounting for their interactions. An instructive example for NLP is shown in figure 5.

Alongside local and global explainability, works like Yeh et al. (2020) adapt the notion of Shapley values for concept analysis (Sajjad et al., 2021). Given a set of concepts extracted from a model, the authors define the notion of *completeness* as a measure to indicate how sufficient such concepts are in explaining the model's predictive behavior. Furthermore, they propose ConceptSHAP, an unsupervised approach able to automatically retrieve a set of interpretable concepts without needing to know them in advance.

5.5 Approaches Proposed for Estimation Efficiency

While Shapley values convey useful information about the importance or contribution of a certain input component, their computation quickly becomes infeasible as coalitions grow exponentially w.r.t. input size. The SHAP framework already addresses this issue by providing more efficient estimation techniques. Nevertheless, later works continued to explore improvements to further decrease the computational effort necessary to produce meaningful explanations.

Chen et al. (2018) leverage features dependencies in image and text data to build two efficient algorithms, *L-Shapley* and *C-Shapley*, for Shapley values estimation. Their methods only consider a subset of the possible coalitions based on the data's underlying graph structure, which connects for instance adjacent words and pixels in texts and images respectively.

SurrogateSHAP (Messalas et al., 2019), instead, trains an XGBoost tree as a surrogate for the original model. The surrogate is then used to generate SHAP explanations, which considerably reduces the computational cost compared to directly applying SHAP to the original (more complex) model.

6 Relevance for NLP Research

Large and complex neural NLP models—such as BERT (Devlin et al., 2019) and GPT-3 (Brown et al., 2020)—are used extensively in research and industry. The trend is justified by the strong correlation between models' size and their performance (Madsen et al., 2021; Brown et al., 2020). Naturally, increasing model complexity causes a higher demand for NLP explainability. In this section, we match this demand to the reviewed SHAP-based methods and provide researchers with use-case-based recommendations.

6.1 Applicability of the Approaches

In table 1 (rightmost column), we also evaluate each SHAP-based explainability approach based on its applicability to neural NLP models. In this regard, our assessment considers *availability of* *implementations*, *suitability for text data*, and *conceptual complexity* as relevant factors. We organize all reviewed approaches into four tiers:

- *Ready Off-the-Shelf*: The code is available and is ready to be used as-is.

- *Adaptable*: The code is available and there are straightforward steps for its adaptation to NLP use cases. Alternatively, no code is available but there are clear instructions for an ad-hoc implementation for the NLP domain.

- *Potentially Applicable*: Strong assumptions and substantial implementation work are required to apply the method to NLP.

- *Not Relevant*: The method is only applicable to other domains and it does not provide any apparent value for explaining NLP models.

6.2 Recommendations for NLP Use Cases

To build feature attribution explanations, HEDGE (Chen et al., 2020) is arguably the most suitable choice, as hierarchical explanations can contain more information than their non-hierarchical counterpart, e.g. generated with SHAP. The strength of HEDGE becomes even more apparent when dealing with long texts, where sentence structure is of major relevance for the model to be explained. *L-Shapley*, *C-Shapley* (Chen et al., 2018) and PartitionSHAP can also be considered where hierarchical explanations are not necessary and very computationally efficient methods are required instead.

For model debugging, Neuron Shapley is suitable to identify neurons that are responsible for unintended biases or that are particularly vulnerable to adversarial attacks (Ghorbani and Zou, 2021). Pruning these neurons can be an effective method of alleviating such model defects (Ghorbani and Zou, 2021). To gain a global understanding of what the model has learned in practice, SAGE (Covert et al., 2020) combined with word grouping provides a summary of the features—e.g. words—that are most relevant for the model's performance. In this case, pruning irrelevant features can be also tested to improve model accuracy. A similar summary can be provided by ConceptSHAP (Yeh et al., 2020), which can compile a comprehensive list of the concepts identified by the model in an unsupervised fashion. Furthermore, ConceptSHAP can be used to determine the amount of model variance covered by the whole set of identified concepts (Yeh et al., 2020).

If causal structures or dependencies present in the text are known and can be explicitly modeled, then methods such as ASV (Frye et al., 2019), Shapley Flow (Wang et al., 2021), and Causal Shapley (Heskes et al., 2020) can leverage such information. For use cases involving graphs as part of multi-modal inputs—e.g. modeling a social network (Wich et al., 2021)—any of the previous methods can be combined with SubGraphX (Yuan et al., 2021) to also produce explanations for the graph component of the input.

When it comes to *sequence-to-sequence* tasks such as question answering and machine translation, the usage of SHAP-based methods has not been explored in depth. With a few exceptions[4], available approaches seem particularly tailored only to classification settings. We believe this is a strong limitation and we encourage the reader to look for alternatives.

7 Criticisms

The usage of Shapley values for generating model explanations has also been criticized. For instance, Kumar et al. (2020) shows that using Shapley values for feature importance leads to mathematical inconsistencies which can only be mitigated by introducing further complexity like causality assumptions. Moreover, the authors argue that Shapley values do not represent an intuitive solution to the human-centric goals of model explanations and thus are only suitable in a limited range of settings.

Sundararajan and Najmi (2020), on the other hand, criticize some Shapley-value-based methods. In fact, while a strong case for utilizing Shapley values can be made thanks to their uniqueness result in satisfying certain properties (see 2.1), often methods employing them operate under different assumptions and hence the uniqueness results loses validity in their context.

Merrick and Taly (2020) argues that existing SHAP-based literature focuses on the axiomatic foundation of Shapley values and their efficient estimation but neglects the uncertainty of the explanations produced. The authors illustrate how small differences in the underlying game formulation can lead to sudden leaps in Shapley values and can attribute a positive contribution to features that do not play any role in the machine learning model.

[4] https://shap.readthedocs.io/en/latest/text_examples.html

8 Conclusion

SHAP is a core contribution to explainable artificial intelligence and one of the most popular frameworks for local interpretability. A considerable amount of recent works has proposed SHAP-based approaches, which we identify as part of five different yet overlapping research directions. In particular, the recent literature has worked towards **(C1)** *tailoring explanations to different input data*, **(C2)** *explaining specific models*, **(C3)** *improving the framework's flexibility via modifying core assumptions*, **(C4)** *producing different explanation types*, and **(C5)** *estimating Shapley values more efficiently*.

This work has reviewed a total of 41 approaches and has organized them based on the introduced categories. As expected, given the overlapping nature of the classification, the majority of existing methods fall into multiple categories and have therefore each made distinct contributions to the field. While most of them are not directly applicable to NLP settings, we identified a few that can be beneficial for current practitioners. Furthermore, we have compiled a list of recommendations for each NLP use case. We also observe a severe limitation of SHAP-based methods in terms of applicability to sequence-to-sequence NLP tasks.

We hope our work provides NLP/XAI practitioners and newcomers with a comprehensive overview of SHAP-based approaches, with references to stimulate further investigation and future advances in academic and industrial research.

Acknowledgments

This paper has been supported by the German *Federal Ministry of Education and Research* (BMBF, grant 01IS17049).

References

Kjersti Aas, Martin Jullum, and Anders Løland. 2021. Explaining individual predictions when features are dependent: More accurate approximations to shapley values. *Artificial Intelligence*, 298:103502.

Marco Ancona, Cengiz Oztireli, and Markus Gross. 2019. Explaining deep neural networks with a polynomial time algorithm for shapley value approximation. In *International Conference on Machine Learning*, pages 272–281. PMLR.

Liat Antwarg, Ronnie Mindlin Miller, Bracha Shapira, and Lior Rokach. 2019. Explaining anomalies detected by autoencoders using shap. *arXiv preprint arXiv:1903.02407*.

Joao Bento, Pedro Saleiro, Andre Cruz, Mario Figueiredo, and Pedro Bizarro. 2021. Timeshap: Explaining recurrent models through sequence perturbations. *KDD*.

Umang Bhatt, Adrian Weller, and José MF Moura. 2020. Evaluating and aggregating feature-based model explanations. *arXiv preprint arXiv:2005.00631*.

Alexander Binder, Sebastian Bach, Gregoire Montavon, Klaus-Robert Müller, and Wojciech Samek. 2016. Layer-wise relevance propagation for deep neural network architectures. In *Information science and applications (ICISA) 2016*, pages 913–922. Springer.

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

Hanjie Chen, Guangtao Zheng, and Yangfeng Ji. 2020. Generating hierarchical explanations on text classification via feature interaction detection. *arXiv preprint arXiv:2004.02015*.

Hugh Chen, Scott Lundberg, and Su-In Lee. 2021. Explaining models by propagating shapley values of local components. In *Explainable AI in Healthcare and Medicine*, pages 261–270. Springer.

Jianbo Chen, Le Song, Martin J Wainwright, and Michael I Jordan. 2018. L-shapley and c-shapley: Efficient model interpretation for structured data. *arXiv preprint arXiv:1808.02610*.

Ian Covert, Scott Lundberg, and Su-In Lee. 2021. Explaining by removing: A unified framework for model explanation. *Journal of Machine Learning Research*, 22(209):1–90.

Ian Covert, Scott M Lundberg, and Su-In Lee. 2020. Understanding global feature contributions with additive importance measures. *Advances in Neural Information Processing Systems*, 33:17212–17223.

Anupam Datta, Shayak Sen, and Yair Zick. 2016. Algorithmic transparency via quantitative input influence: Theory and experiments with learning systems. In *2016 IEEE symposium on security and privacy (SP)*, pages 598–617.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *NAACL-HLT (1)*.

Kedar Dhamdhere, Ashish Agarwal, and Mukund Sundararajan. 2020. The shapley taylor interaction index. *PMLR*.

Christopher Frye, Colin Rowat, and Ilya Feige. 2019. Asymmetric shapley values: incorporating causal knowledge into model-agnostic explainability. *NeurIPS 2020*.

Amirata Ghorbani and James Zou. 2019. Data shapley: Equitable valuation of data for machine learning. In *International Conference on Machine Learning*, pages 2242–2251. PMLR.

Amirata Ghorbani and James Zou. 2021. Neuron shapley: Discovering the responsible neurons. *NeurIPS 2021*.

Tom Heskes, Evi Sijben, Ioan Gabriel Bucur, and Tom Claassen. 2020. Causal shapley values: Exploiting causal knowledge to explain individual predictions of complex models. *NeurIPS 2020*.

Lukas Huber, Marc Alexander Kühn, Edoardo Mosca, and Georg Groh. 2022. Detecting word-level adversarial text attacks via SHapley additive exPlanations. In *Proceedings of the 7th Workshop on Representation Learning for NLP*, pages 156–166, Dublin, Ireland. Association for Computational Linguistics.

Joseph Janizek, Pascal Sturmfels, and Su-In Lee. 2021. Explaining explanations: Axiomatic feature interactions for deep networks. *JMLR*.

I Elizabeth Kumar, Suresh Venkatasubramanian, Carlos Scheidegger, and Sorelle Friedler. 2020. Problems with shapley-value-based explanations as feature importance measures. In *International Conference on Machine Learning*, pages 5491–5500. PMLR.

Indra Kumar, Carlos Scheidegger, Suresh Venkatasubramanian, and Sorelle Friedler. 2021. Shapley residuals: Quantifying the limits of the shapley value for explanations. *NeurIPS*.

Christophe Labreuche and Simon Fossier. 2018. Explaining multi-criteria decision aiding models with an extended shapley value. In *IJCAI*, pages 331–339.

Pantelis Linardatos, Vasilis Papastefanopoulos, and Sotiris Kotsiantis. 2021. Explainable ai: A review of machine learning interpretability methods. *Entropy*, 23(1):18.

Stan Lipovetsky and Michael Conklin. 2001. Analysis of regression in game theory approach. *Applied Stochastic Models in Business and Industry*, 17(4):319–330.

Scott M. Lundberg, Gabriel Erion, Hugh Chen, Alex DeGrave, Jordan M. Prutkin, Bala Nair, Ronit Katz, Jonathan Himmelfarb, Nisha Bansal, and Su-In Lee. 2020a. From local explanations to global understanding with explainable ai for trees. *Nature Machine Intelligence*, 2(1):2522–5839.

Scott M Lundberg, Gabriel Erion, Hugh Chen, Alex DeGrave, Jordan M Prutkin, Bala Nair, Ronit Katz, Jonathan Himmelfarb, Nisha Bansal, and Su-In Lee. 2020b. From local explanations to global understanding with explainable ai for trees. *Nature machine intelligence*, 2(1):56–67.

Scott M Lundberg, Gabriel G Erion, and Su-In Lee. 2018. Consistent individualized feature attribution for tree ensembles. *arXiv preprint arXiv:1802.03888*.

Scott M. Lundberg and Su-In Lee. 2017. A unified approach to interpreting model predictions. *NeurIPS 2017*.

Andreas Madsen, Siva Reddy, and Sarath Chandar. 2021. Post-hoc interpretability for neural nlp: A survey. *arXiv preprint arXiv:2108.04840*.

Luke Merrick and Ankur Taly. 2020. The explanation game: Explaining machine learning models using shapley values. In *International Cross-Domain Conference for Machine Learning and Knowledge Extraction*, pages 17–38. Springer.

Andreas Messalas, Yiannis Kanellopoulos, and Christos Makris. 2019. Model-agnostic interpretability with shapley values. In *2019 10th International Conference on Information, Intelligence, Systems and Applications (IISA)*, pages 1–7. IEEE.

Edoardo Mosca, Defne Demirtürk, Luca Mülln, Fabio Raffagnato, and Georg Groh. 2022a. GrammarSHAP: An efficient model-agnostic and structure-aware NLP explainer. In *Proceedings of the First Workshop on Learning with Natural Language Supervision*, pages 10–16, Dublin, Ireland. Association for Computational Linguistics.

Edoardo Mosca, Katharina Harmann, Tobias Eder, and Georg Groh. 2022b. Explaining neural NLP models for the joint analysis of open-and-closed-ended survey answers. In *Proceedings of the 2nd Workshop on Trustworthy Natural Language Processing (TrustNLP 2022)*, pages 49–63, Seattle, U.S.A. Association for Computational Linguistics.

Edoardo Mosca, Maximilian Wich, and Georg Groh. 2021. Understanding and interpreting the impact of user context in hate speech detection. In *Proceedings of the Ninth International Workshop on Natural Language Processing for Social Media*, pages 91–102.

Md Rizwan Parvez and Kai-Wei Chang. 2021. Evaluating the values of sources in transfer learning. *NAACL 2021*.

Yanou Ramon, David Martens, Foster Provost, and Theodoros Evgeniou. 2019. Counterfactual explanation algorithms for behavioral and textual data. *arXiv preprint arXiv:1912.01819*.

Nickalus Redell. 2019. Shapley decomposition of r-squared in machine learning models. *arXiv preprint arXiv:1908.09718*.

Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. 2016. " why should i trust you?" explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, pages 1135–1144.

Hassan Sajjad, Narine Kokhlikyan, Fahim Dalvi, and Nadir Durrani. 2021. Fine-grained interpretation and causation analysis in deep nlp models. *arXiv preprint arXiv:2105.08039*.

Lloyd S Shapley. 1953. A value for n-person games. *Contributions to the Theory of Games 2.28*, page 307–317.

Avanti Shrikumar, Peyton Greenside, and Anshul Kundaje. 2017. Learning important features through propagating activation differences. In *International Conference on Machine Learning*, pages 3145–3153. PMLR.

Raghav Singal, Omar Besbes, Antoine Desir, Vineet Goyal, and Garud Iyengar. 2019. Shapley meets uniform: An axiomatic framework for attribution in online advertising. In *The World Wide Web Conference*, pages 1713–1723.

Eunhye Song, Barry L Nelson, and Jeremy Staum. 2016. Shapley effects for global sensitivity analysis: Theory and computation. *SIAM/ASA Journal on Uncertainty Quantification*, 4(1):1060–1083.

Erik Štrumbelj and Igor Kononenko. 2014. Explaining prediction models and individual predictions with feature contributions. *Knowledge and information systems*, 41(3):647–665.

Mukund Sundararajan and Amir Najmi. 2020. The many shapley values for model explanation. In *International Conference on Machine Learning*, pages 9269–9278. PMLR.

Mukund Sundararajan, Ankur Taly, and Qiqi Yan. 2017. Axiomatic attribution for deep networks. In *International Conference on Machine Learning*, pages 3319–3328. PMLR.

Sarah Tan, Giles Hooker, Paul Koch, Albert Gordo, and Rich Caruana. 2018. Considerations when learning additive explanations for black-box models. *arXiv preprint arXiv:1801.08640 3*.

Jacopo Teneggi, Alexandre Luster, and Jeremias Sulam. 2021. Fast hierarchical games for image explanations. *arXiv preprint arXiv:2104.06164*.

Giulia Vilone and Luca Longo. 2020. Explainable artificial intelligence: a systematic review. *arXiv preprint arXiv:2006.00093*.

Guan Wang. 2019. Interpret federated learning with shapley values. *arXiv preprint arXiv:1905.04519*.

Jianhong Wang, Yuan Zhang, Tae-Kyun Kim, and Yunjie Gu. 2020. Shapley q-value: A local reward approach to solve global reward games. *AAAI*, 34:7285–7292.

Jiaxuan Wang, Jenna Wiens, and Scott Lundberg. 2021. Shapley flow: A graph-based approach to interpreting model predictions. *AISTATS 2021*.

Maximilian Wich, Edoardo Mosca, Adrian Gorniak, Johannes Hingerl, and Georg Groh. 2021. Explainable abusive language classification leveraging user and network data. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 481–496. Springer.

Brian Williamson and Jean Feng. 2020. Efficient nonparametric statistical inference on population feature importance using shapley values. In *International Conference on Machine Learning*, pages 10282–10291. PMLR.

Chih-Kuan Yeh, Been Kim, Sercan Arik, Chun-Liang Li, Tomas Pfister, and Pradeep Ravikumar. 2020. On completeness-aware concept-based explanations in deep neural networks. *Advances in Neural Information Processing Systems*, 33.

Hao Yuan, Haiyang Yu, Jie Wang, Kang Li, and Shuiwang Ji. 2021. On explainability of graph neural networks via subgraph explorations. *arXiv preprint arXiv:2102.05152*.

A.2 STUDY III

Edoardo Mosca, Maximilian Wich, and Georg Groh (June 2021). “Understanding and Interpreting the Impact of User Context in Hate Speech Detection.” In: *Proceedings of the Ninth International Workshop on Natural Language Processing for Social Media*. Online: Association for Computational Linguistics, pp. 91–102. DOI: `10.18653/v1/2021.socialnlp-1.8`. URL: `https://aclanthology.org/2021.socialnlp-1.8`

[2] `https://creativecommons.org/licenses/by/4.0/`

Publication Summary

"As hate speech spreads on social media and online communities, research continues to work on its automatic detection. Recently, recognition performance has been increasing thanks to advances in deep learning and the integration of user features. This work investigates the effects that such features can have on a detection model. Unlike previous research, we show that simple performance comparison does not expose the full impact of including contextual- and user information. By leveraging explainability techniques, we show (1) that user features play a role in the model's decision and (2) how they affect the feature space learned by the model. Besides revealing that—and also illustrating why—user features are the reason for performance gains, we show how such techniques can be combined to better understand the model and to detect unintended bias." (Mosca, Wich, and Groh, 2021, p. 1)

Author Contributions

Edoardo Mosca contributed to the study as follows:

- Conception, development, and lead of the research project **70%**
- Literature review and feasibility study **70%**
- Methodology and experimental design **80%**
- Implementation and interpretation of results. **80%**
- Drafting of the manuscript **80%**
- Submission, peer review, and publication process **80%**

Understanding and Interpreting the Impact of User Context in Hate Speech Detection

Edoardo Mosca
TU Munich,
Department of Informatics,
Germany
edoardo.mosca@tum.de

Maximilian Wich
TU Munich,
Department of Informatics,
Germany
maximilian.wich@tum.de

Georg Groh
TU Munich,
Department of Informatics,
Germany
grohg@in.tum.de

Abstract

As hate speech spreads on social media and online communities, research continues to work on its automatic detection. Recently, recognition performance has been increasing thanks to advances in deep learning and the integration of user features. This work investigates the effects that such features can have on a detection model. Unlike previous research, we show that simple performance comparison does not expose the full impact of including contextual- and user information. By leveraging explainability techniques, we show (1) that user features play a role in the model's decision and (2) how they affect the feature space learned by the model. Besides revealing that—and also illustrating *why*—user features are the reason for performance gains, we show how such techniques can be combined to better understand the model and to detect unintended bias.

1 Introduction

Communication and information exchange between people is taking place on online platforms at a continuously increasing rate. While these means allow everyone to express themselves freely at any time, they are massively contributing to the spread of negative phenomenons such as online harassment and abusive behavior. Among those, which are all to discourage, online hate speech has attracted the attention of many researchers due to its deleterious effects (Munro, 2011; Williams et al., 2020; Duggan, 2017).

The extremely large volume of online content and the high speed at which new one is generated exclude immediately the chance of content moderation being done manually. This realization has naturally captured the attention of the *Machine Learning* (ML) field, seeking to craft automatic and scalable solutions (MacAvaney et al., 2019; Waseem et al., 2017; Davidson et al., 2017).

Methods for detecting hate speech and similar abusive behavior have been thus on the rise, consistently improving in terms of performance and generalization (Schmidt and Wiegand, 2017; Mishra et al., 2019b). However, even the current state of the art still faces limitations in accuracy and is yet not ready to be deployed in practice. Hate speech recognition remains an extremely difficult task (Waseem et al., 2017), in particular when the expression of hate is implicit and hidden behind figures of speech and sarcasm.

Alongside language features, recent works have considered utilizing user features as an additional source of knowledge to provide detection models with context information (Fehn Unsvåg and Gambäck, 2018; Ribeiro et al., 2018). As a general trend, models incorporating context exhibit improved performance compared to their pure text-based counterparts (Mishra et al., 2018, 2019a). Nevertheless, the effect, which these additional features have on the model, has not been interpreted or understood yet. So far, models have mostly been compared only in terms of performance metrics. The goal of this work is to shed light on the impact generated by including user features—or more in general context—into hate speech detection methods. Our methodology heavily relies on a combination of modern techniques coming from the field of *eXplainable Artificial Intelligence* (XAI).

We show that adding user and social context to models is the reason for performance gains. We also explore the model's learned features space to understand how such features are leveraged for detection. At the same time, we discover that models incorporating user features suffer less from bias in the text. Unfortunately, those same models contain a new type of bias that originates from adding user information.

Proceedings of the Ninth International Workshop on Natural Language Processing for Social Media, pages 91–102
Online Workshop, June 10, 2021. ©2021 Association for Computational Linguistics

2 Related Work

2.1 Explainability for Recognition Models

A limited amount of research has focused on applying XAI techniques to the hate speech recognition case. For instance, Wang (2018) adapts a number of explainability techniques from the computer vision and applies them to a hate speech classifier trained on Davidson et al. (2017). Feature occlusion was used to highlight the most relevant words for the final classifier prediction and activation maximization selected the terms that the classifier captured and judged as relevant at a dataset-level. Vijayaraghavan et al. (2019) constructs an interpretable multi-modal detector that uses text alongside social and cultural context features. The authors leverage attention scores to quantify the relevance of different input features. Wich et al. (2020) applies post-hoc explainability on a custom dataset in German to expose and estimate the impact of political bias on hate speech classifiers. More in detail, left- and right-wing political bias within the training data is visualized via DeepSHAP-based explanations (Lundberg and Lee, 2017).

MacAvaney et al. (2019) combines together multiple simple classifiers to assemble a transparent model. Risch et al. (2020) reviews and compares several explainability techniques applied to hate speech classifiers. Their experimentation includes popular post-hoc approaches such as LIME (Ribeiro et al., 2016) and LRP (Bach et al., 2015) as well as self-explanatory detectors (Risch et al., 2020).

For our use case, we apply *post-hoc explainability* approaches (Lipton, 2018). We use external techniques to explain models that would otherwise be black-boxes (Arrieta et al., 2020). In contrast, *transparent models* are interpretable thanks to their intuitive and simple design.

2.2 Context Features for Hate Speech Detection

Models have been continuously improving since the first documented step towards automatic hate speech detection Spertus (1997). The evolution of recognition approaches has been favored by advances in *Natural Language Processing* (NLP) research (Mishra et al., 2019b). For instance, s.o.t.a detectors like Mozafari et al. (2020) exploit high-performing language models such as BERT (Devlin et al., 2019).

A different research branch took an alternative path and explored the inclusion of social context alongside text. These additional features are usually referred to with the terms *user features*, *context features*, or *social features*. Some tried incorporating the gender (Waseem, 2016) and the profile's geolocation and language (Galán-García et al., 2016). Others instead utilized the user's number of followers or friends (Fehn Unsvåg and Gambäck, 2018).

Modeling users' social and conversational interactions via their corresponding graph was also shown to be rewarding (Mishra et al., 2019b; Cecillon et al., 2019). Ribeiro et al. (2018) creates additional features by measuring properties like betweenness and eigenvector centrality. Mishra et al. (2018) and Mishra et al. (2019a) instead fed the graph directly to the model either embedded as matrix or via using graph convolutional neural network (Hamilton et al., 2017).

While previous work explored the usage of a wide range of context features (Fehn Unsvåg and Gambäck, 2018), detection models have only been compared in terms of performance metrics. Besides accuracy, researchers have not focused on other changes that such features could have on the model. Our work shows that indeed this addition entails a large impact on the recognition algorithm's behavior and substantially changes its characteristics.

3 Experimental Setup

In this section, we describe in detail the different datasets and detection models that we include in our interpretability-driven analysis.

3.1 Data and Preprocessing

Previous research has produced several datasets to support further developments in the hate speech detection area (Founta et al., 2018; Warner and Hirschberg, 2012). Some became relatively popular to benchmark and test new ideas and improvements in recognition techniques. For our experimentation, we pick the DAVIDSON (Davidson et al., 2017) and the WASEEM (Waseem and Hovy, 2016) datasets. The choice was motivated by their variety of speech classes and popularity as detection benchmarks.

Both benchmarks consist of a collection of tweets coupled with classification tasks with three possible classes. DAVIDSON contains $\sim 25,000$ tweets of which $1,430$ are labeled as *hate*, $19,190$ as *offensive*, and $4,163$ as *neither* (Davidson et al., 2017). As classification outcomes in WASEEM instead, we have *racism*, *sexism*, and *neither*. The three classes contain $3,378$, $1,970$, and $11,501$ tweets respectively (Waseem and Hovy, 2016). We were not able to retrieve the remaining 65 of the original $16,914$ samples.

We follow the same preprocessing steps for both datasets. First, terms belonging to categories like *url, email, percent, number, user,* and *time* are annotated via a category token. For instance, "*341*" is replaced by "*<number>*". After that, we apply word segmentation and spell correction based on Twitter word statistics. Both methods and statistics were provided by the *ekphrasis* [1] text preprocessing tool (Baziotis et al., 2017).

In addition to the tweets that represent the text (or content) component of our input features, we also retrieve information about the tweet's authors and their relationships. In a similar fashion as done in Mishra et al. (2018), we construct a *community graph* $G = (V, E)$ where each node represents a user and two nodes are connected if at least one of the two users follows the other one. We were able to retrieve $|V| = 6,725$ users and $|E| = 19,597$ relationships for DAVIDSON, while for WASEEM we have $|V| = 2,024$ and $|E| = 9,955$.

The respective average node degrees are $2,914$ and $4,918$ and the overall graphs' densities:

$$D = \frac{2 \cdot |E|}{|V|(|V| - 1)}$$

are 0.00087 and 0.00486 respectively.

We immediately notice that both graphs are very sparse. In particular, we have $3,393$ users not connected to anyone in DAVIDSON and 927 in WASEEM. For reference, Mishra et al. (2018) achieves a graph density of 0.0075 on WASEEM, with only ~ 400 authors being solitary, i.e. with no connections. We assume the difference is reasonable as data availability considerably decreases over time.

3.2 Detection Models

Our experimentation and findings are based on the comparison of two detection models, one that solely relies on text features and one that instead incorporates context features. To better capture their behavioral differences, we build them to be relatively simple and also to not differ in the text-processing part.

The first model, shown in figure 1, computes the three classification probabilities only based on the tweets' content. The input text is fed to the model as *Bag of Words* (BoW), which is then processed by two fully connected layers. We refer to this model as *text model*.

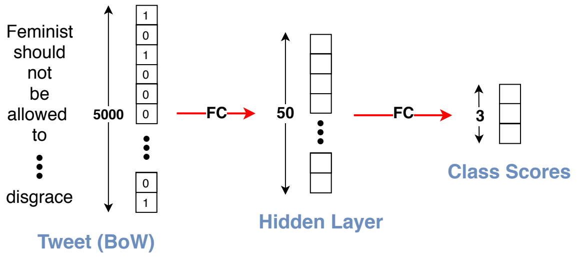


Figure 1: Architecture of the text model.

The second model instead leverages the information coming from three input sources: the tweet's text, the user's vocabulary, and the follower network. The first input is identical to what is fed to the text model. The second is constructed from all the tweets of the author in the dataset and aims to model their overall writing style. Concretely, we merge the tweets' BoW representations, i.e. we apply a logical-OR to their corresponding vectors. The third is the author's follower network and describes their online surrounding community. On a more technical note, this can be extracted as a row from the adjacency matrix of our community graph described in section 3.1. Note that s.o.t.a hate speech detector used similar context features (Mishra et al., 2018, 2019a). We refer to this model as *social model*.

As sketched in figure 2, the different input sources are initially processed separately in the model's architecture. After the first layer, the intermediate representations from the different branches are concatenated together and fed to two more layers to compute the final output. Note that the text- and social models have the same dimensions for their final hidden layer and can be seen as equivalent networks working on different inputs.

4 Proposed Analysis

We now describe our methodology in detail. Recall that our models differ precisely on the usage of user features. As we will see shortly, their comparison beyond accuracy measurements sheds light on the different model properties and hence on the potential impact of incorporating context features.

[1] https://github.com/cbaziotis/ekphrasis

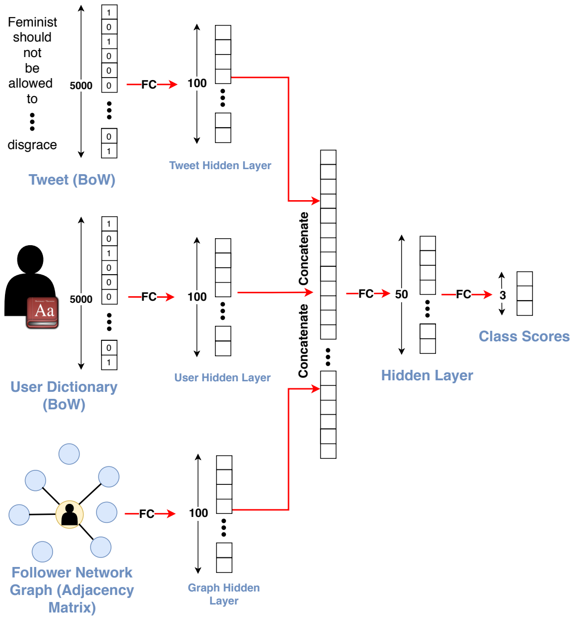


Figure 2: Architecture of the social model.

4.1 Training and Performance

We apply the same training and testing procedure to all models and datasets. We keep the 60% of the data for training while splitting the remaining equally between validation and test set, i.e. 20% each.

Tables 1 and 2 report our results in terms of F1 scores for Waseem (Waseem and Hovy, 2016) and Davidson (Davidson et al., 2017) respectively. To increase our confidence in their validity, we average the performance over five runs with randomly picked train/validation/test sets. We observe different trends for the two datasets.

Speech Class	Text Model	Social Model
Racism	0.711	0.735
Sexism	0.703	0.832
Neither	0.881	0.907
Overall	0.829	0.872

Table 1: F1 Scores on Waseem and Hovy (2016).

On Waseem, the social model considerably outperforms (by 4.3%) our text model. The performance gain is general and not restricted to any single class. Quite surprisingly, our text model performs better on racist tweets than sexist ones, although the sexism class is almost twice as big. This suggests that sexism is, at least in this case, somewhat harder to detect by just looking at the tweet content. On the contrary, our social model shows an impressive improvement in the sexism class (almost 13%), suggesting the presence of detectable patterns in sexist users and their social interactions.

Speech Class	Text Model	Social Model
Hate	0.154	0.347
Offensive	0.939	0.939
Neither	0.809	0.815
Overall	0.876	0.886

Table 2: F1 Scores on Davidson et al. (2017).

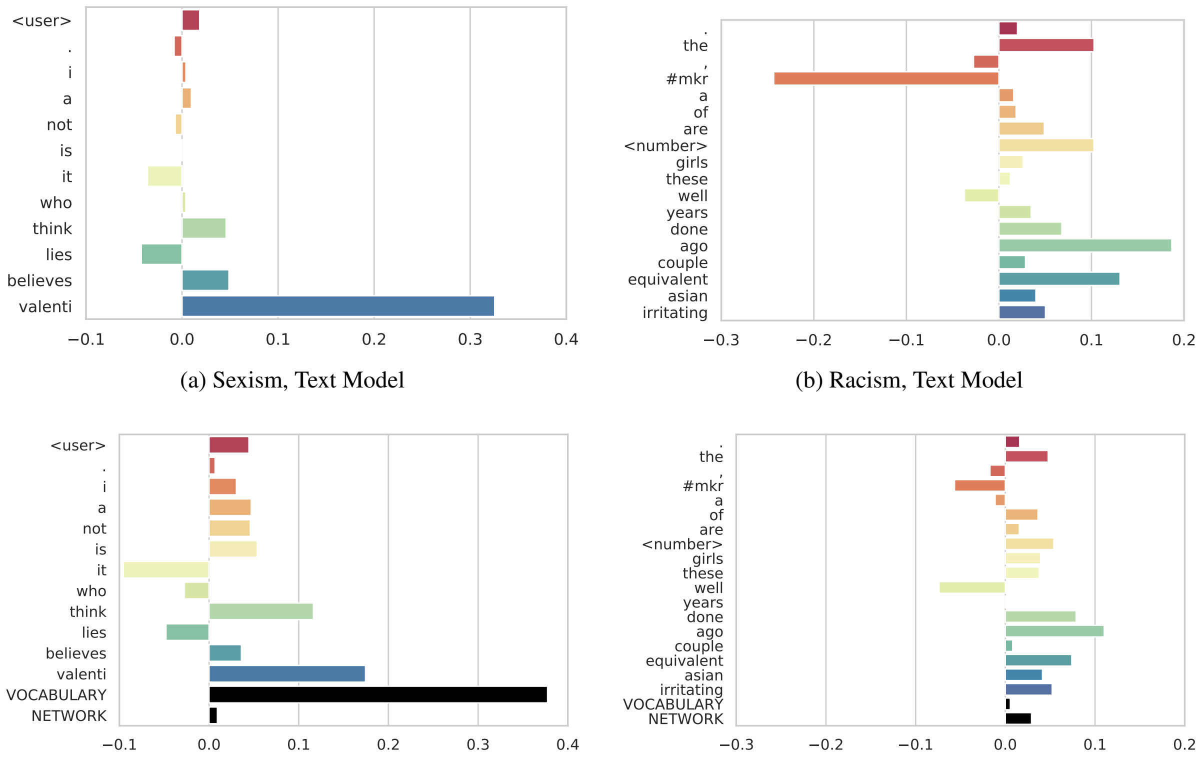
On Davidson, we only observe a contained improvement (1%). Moreover, the jump in performance is restricted to the hate class, containing a tiny amount of samples. We believe the difference between the two datasets should be expected due to the lower amount of user data available for Davidson. Considering these results, we focus on applying our technique on the Waseem dataset in the remainder of this paper. Nevertheless, the respective results on Davidson can be found in the appendix A. While on both datasets we do not outperform the current s.o.t.a—Mishra et al. (2019a) on Waseem and Mozafari et al. (2020) on Davidson—our results are comparable and thus satisfactory for our purposes.

4.2 Shapley Values Estimation

We now apply a first post-hoc explainability method. For each feature we calculate its corresponding *Shapley value* (Shapley, 1953; Lundberg and Lee, 2017). That is, we quantify the relevance that each feature has for the prediction of a specific output. Shapley values have been shown—both theoretically and empirically—to be an ideal estimator for feature relevance (Lundberg and Lee, 2017).

As exact Shapley values are exponentially complex to determine, we use accurate approximation methods as done in (Lundberg and Lee, 2017; Štrumbelj and Kononenko, 2014). Figure 3 shows concrete examples in which Shapley values are calculated for both models on two test tweets from Waseem.

For our social model, we consider the user vocabulary and the follower network as single features for simplicity. Notably, the context is used by the social model and can play a significant role in its prediction. Hence, we can confirm the context features to be the reason for the performance gains. We can empirically exclude that the differences between the text- and the social model architectures justify the jump in performance.



(a) Sexism, Text Model

(b) Racism, Text Model

(c) Sexism, Social Model

(d) Racism, Social Model

Figure 3: Example of features contribution, computed via Shapley value approximation, for our text and social models. In (a) and (c) we use as input the tweet "*<user> I think Arquette is a dummy who believes it. Not a Valenti who knowingly lies.*". The sexist tweet refers to the actress Patricia Arquette, who spoke in favour of gender equality, and the feminist writer Jessica Valenti. Some words are missing in the plot as our BoW dimension is limited during preprocessing. In (b) and (d), we use the racist tweet "*These girls are the equivalent of the irritating Asian girls a couple of years ago. Well done, 7. #MKR*". The hashtag refers to the Australian cooking show "*My Kitchen Rules*".

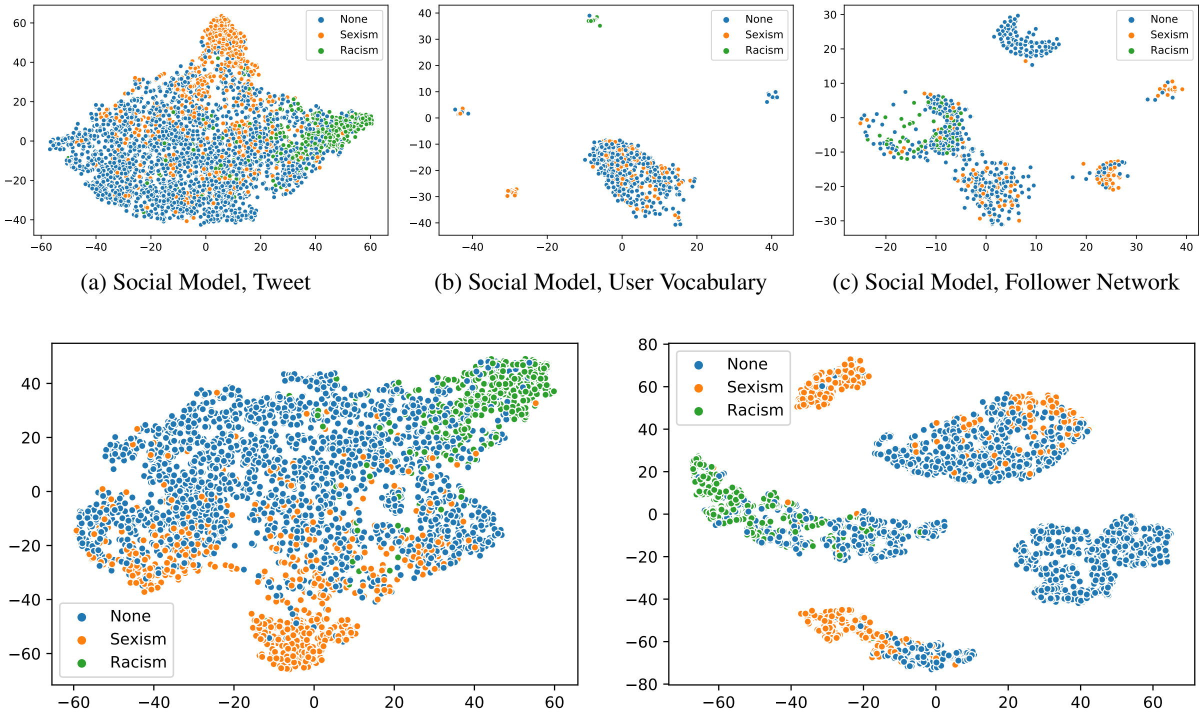
4.3 Feature Space Exploration

We have seen that detection models can benefit from the inclusion of context features. We now focus on understanding *why* this is the case. Shapley values and more in general feature attribution methods can quantify *how much* single features contribute to the prediction. Yet, alone, they do not give us any intuition to answer our why-question.

We look at the feature space learned by our models, which can be considered a global explainability technique. For our text model, we remove the last layer and feed the tweets to the remaining architecture. The output is a 50-dimensional embedding for each tweet. We employ the *t-Distributed Stochastic Neighbor Embedding* (t-SNE) (Van der Maaten and Hinton, 2008) to reduce the embeddings to two dimensions for visualization purposes.

The resulting plot, in figure 4d, shows all the tweets in a single cluster. Racist tweets look more concentrated in one area than sexist ones, suggesting that sexism is somewhat harder to detect for the model. This result is coherent with our per-class performance scores.

We apply the same procedure to the social model. In this case, we visualize the hidden layer of each separate branch as well as the final hidden layer analogous to the text model. Not surprisingly, the tweet branch (figure 4a) looks very similar to the feature space learned by our text model. The user's vocabulary branch (figure 4b) instead shows the samples distributed in well-separated clusters. Notably, racist tweets have been restricted to one cluster and we can also observe pure-sexist and pure-neither clusters. The follower network branch (figure 4c) looks similar though cluster separation is not as strong. Once more, we notice racism more concentrated than sexism, which is considerably more mixed with regular tweets. To some extent, this result is in line with the notion of *homophily* among racist users (Mathew et al., 2019).



(a) Social Model, Tweet (b) Social Model, User Vocabulary (c) Social Model, Follower Network

(d) Text Model (e) Social Model

Figure 4: WASEEM tweets, colored by label, in the features space learned by our text model (d) and social model (a,b,c for the independent branches, e combined).

Intuitively, being able to divide users into different clusters based on their behavior should be helpful for classification at later layers. This is confirmed by the combined feature space plot (figure 4e). Indeed, tweets are now structured in multiple clusters instead of a single one as for our text model. Also in this case, we observe several pure or almost-pure groups.

The corresponding visualizations and results for DAVIDSON can be found in appendix A.

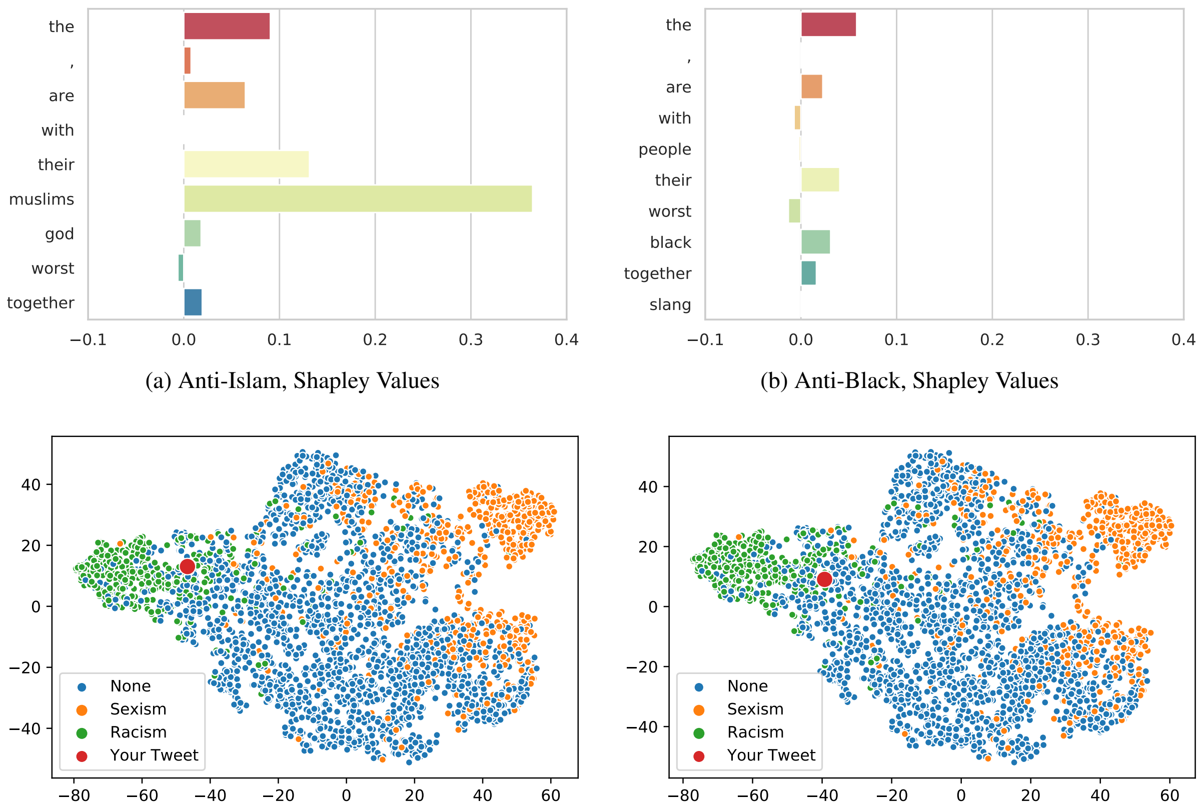
4.4 Targeted Behavioral Analysis: Explaining a Novel Tweet

We have seen how different explainability techniques convey different types of information on the examined model. Computing Shapley values and visualizing the learned feature space can also be used in combination as they complement each other. If used together, they can both quantify the relevance of each feature as well as show how certain types of features are leveraged by the model to better distinguish between classes.

So far, our explanations are relative to the datasets used for model training and testing. However, to better understand a classifier it should also be tested beyond its test set. This can be simply done by feeding the model with a novel tweet. Via artificially crafting tweets, we can check the model's behavior in specific cases. For instance, we can inspect how it reacts to specific sub-types of hate.

Let us consider the anti-Islamic tweet "*muslims are the worst, together with their god*". If fed to our model, it is classified as racist with a 75% confidence following our expectations. Figures 5a and 5c show explanations for the tweet. We can see that the word "*muslim*" plays a big role by looking at its corresponding Shapley value. At the same time, the projection of the novel tweet onto the feature space shows how the sample is collocated together with the other racist tweets by the text model.

If we now change our hypothetical tweet to be anti-black—"*black people are the worst, together with their slang*"—we observe a different model behavior (figures 5b and 5d). In fact, now the tweet is not classified as racist. No word has a substantial impact on the prediction. We can also notice a slight shift of the sample in the features space, away from the racism cluster. If changing the target of the hate changes the prediction, then the model/dataset probably contains bias against that target. Model interpretability further reveals how



(a) Anti-Islam, Shapley Values

(b) Anti-Black, Shapley Values

(c) Anti-Islam, Embedding in Latent Space

(d) Anti-Black, Embedding in Latent Space

Figure 5: Features contribution (Shapley values w.r.t. the racism class) and embedding in the text model's latent space of an islamophobic and a anti-black racist tweets. The two sentences had, according to our text model, the 75% and 24% probability of being racist respectively.

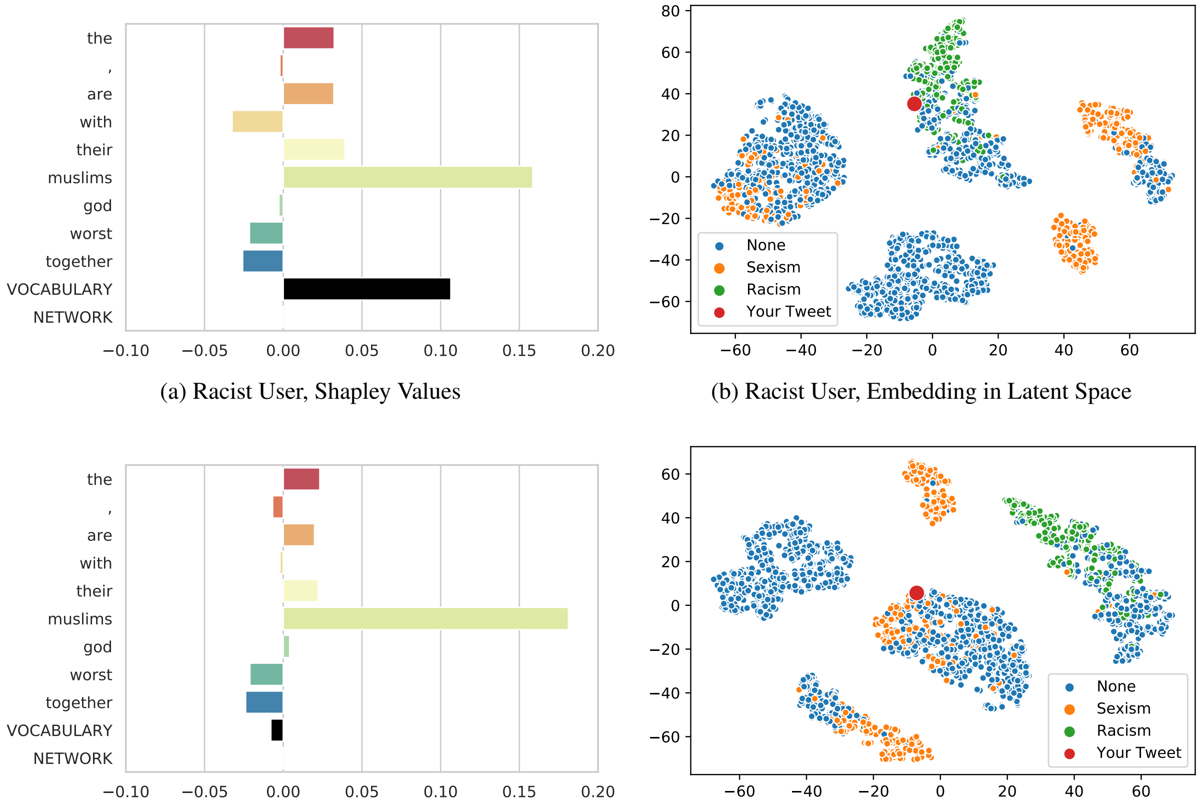
its behavior reacts to different targets.

We run the same experiment with our social model. This time, it correctly classifies the anti-black tweet as racist (55% confidence). This suggests that text bias could be mitigated by using models that do not only rely on the text input. However, the social model is much more sensitive to changes in the user-derived features. To test this, we feed the model the same tweet and only change the author that generated it. For a fair comparison, we pick one random user with other racist tweets, one random user with other sexist tweets, and one random user with no hateful tweets in the dataset. We refer to these users as racist, sexist, and regular users respectively.

Our crafted tweet is classified as racist when coming from a racist user (64%). However, it is instead judged non-hateful in both the other cases (12% and 19% for a sexist and user with no hate background respectively). Evidently, racist tweets also need some contribution from the social features to be judged as racist.

A very informative explanation comes again from both the Shapley values and the feature space exploration (figure 6). On the left side, we can see the Shapley value for the racist and regular users. Results relative to the sexist user are analogous to the regular user and reported in the supplementary material (A.3). All the words have a similar contribution to the racism class in all cases. However, the difference in the authors plays a substantial role in the decision. Only the racist user positively contributes to the racism class. On the right side of 6, we can see the embedding in the latent space for each case. Different input authors cause the tweet to be embedded in different clusters. Only in the first one the model actually considers the possibility of the tweet being racist.

Hence, while adding user-derived features might mitigate the effects of bias in the text, it generates a new form of bias that could discriminate users based on their previous behavior and hinder the model from classifying correctly hateful content.



(a) Racist User, Shapley Values

(b) Racist User, Embedding in Latent Space

(c) Regular User, Shapley Values

(d) Regular User, Embedding in Latent Space

Figure 6: Features contribution (w.r.t. racism class) and embeddings of the islamophobic tweet in the social model's latent space. The two pairs of plots are w.r.t. two predictions done with different users as input: a racist one (a,b, 64%), and a regular one (c,d, 19%).

5 Conclusion and Future Work

In our work, we investigated the effects of user features in hate speech detection. In previous studies, this was done by comparing models based on performance metric. We have shown that post-hoc explainability techniques provide a much deeper understanding of the models' behavior. In our case, when applied to two models that differ specifically on the usage of context features, the in-depth comparison reveals the impact that such additional features can have.

The two utilized techniques—*Shapley values estimation* and *learned feature space exploration*—convey different kinds of information. The first one quantifies how each feature plays a role but does not tell us what is happening in the background. The second one illustrates the model's perception of the tweets but does not provide any quantitative information for the prediction. Furthermore, we have seen that artificially crafting and modifying a tweet can be useful to examine the models' behavior in particular scenarios. In concrete examples, the two approaches worked as bias detectors present in the text as well as in the user features.

We believe that analyzing detection models is vital for understanding how certain features shape the way data is processed. Accuracy alone is by no means a sufficient metric to decide which model to prefer. Our work shows that even models that perform significantly better can potentially lead to new types of bias. We urge researchers in the field to compare recognition approaches beyond accuracy to avoid potential harm to affected users.

Data scarcity is still a main issue faced by current researchers, especially when it comes to context features. We believe that larger and more complete datasets will improve our understanding of how certain features interact and will help future research in advancing both in accuracy and bias mitigation.

Acknowledgments

This paper is based on a joined work in the context of Edoardo Mosca's master's thesis (Mosca, 2020).

References

Alejandro Barredo Arrieta, Natalia Díaz-Rodríguez, Javier Del Ser, Adrien Bennetot, Siham Tabik, Alberto Barbado, Salvador García, Sergio Gil-López, Daniel Molina, Richard Benjamins, et al. 2020. Explainable Artificial Intelligence (XAI): Concepts, taxonomies, opportunities and challenges toward responsible AI. *Information Fusion*, 58:82–115.

Sebastian Bach, Alexander Binder, Grégoire Montavon, Frederick Klauschen, Klaus-Robert Müller, and Wojciech Samek. 2015. On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation. *PloS one*, 10(7).

Christos Baziotis, Nikos Pelekis, and Christos Doulkeridis. 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.

Noé Cecillon, Vincent Labatut, Richard Dufour, and Georges Linarès. 2019. Abusive language detection in online conversations by combining content- and graph-based features. In *ICWSM International Workshop on Modeling and Mining Social-Media-Driven Complex Networks*, volume 2, page 8. Frontiers.

Thomas Davidson, Dana Warmsley, Michael Macy, and Ingmar Weber. 2017. Automated hate speech detection and the problem of offensive language. In *Proceedings of the International AAAI Conference on Web and Social Media*, volume 11.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Maeve Duggan. 2017. *Online harassment 2017*. Pew Research Center.

Elise Fehn Unsvåg and Björn Gambäck. 2018. The Effects of User Features on Twitter Hate Speech Detection. In *Proc. 2nd Workshop on Abusive Language Online*, pages 75–85.

Antigoni-Maria Founta, Constantinos Djouvas, Despoina Chatzakou, Ilias Leontiadis, Jeremy Blackburn, Gianluca Stringhini, Athena Vakali, Michael Sirivianos, and Nicolas Kourtellis. 2018. Large scale crowdsourcing and characterization of twitter abusive behavior. In *Proc. 11th ICWSM*, pages 491–500.

Patxi Galán-García, José Gaviria de la Puerta, Carlos Laorden Gómez, Igor Santos, and Pablo García Bringas. 2016. Supervised machine learning for the detection of troll profiles in twitter social network: Application to a real case of cyberbullying. *Logic Journal of the IGPL*, 24(1):42–53.

Will Hamilton, Zhitao Ying, and Jure Leskovec. 2017. Inductive representation learning on large graphs. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

Zachary C Lipton. 2018. The mythos of model interpretability. *Queue*, 16(3):31–57.

Scott M Lundberg and Su-In Lee. 2017. A unified approach to interpreting model predictions. In *Advances in neural information processing systems*, pages 4765–4774.

Laurens Van der Maaten and Geoffrey Hinton. 2008. Visualizing data using t-sne. *Journal of machine learning research*, 9(11).

Sean MacAvaney, Hao-Ren Yao, Eugene Yang, Katina Russell, Nazli Goharian, and Ophir Frieder. 2019. Hate speech detection: Challenges and solutions. *PloS one*, 14(8).

Binny Mathew, Ritam Dutt, Pawan Goyal, and Animesh Mukherjee. 2019. Spread of hate speech in online social media. In *Proceedings of the 10th ACM conference on web science*, pages 173–182.

Pushkar Mishra, Marco Del Tredici, Helen Yannakoudakis, and Ekaterina Shutova. 2018. Author profiling for abuse detection. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 1088–1098.

Pushkar Mishra, Marco Del Tredici, Helen Yannakoudakis, and Ekaterina Shutova. 2019a. Abusive language detection with graph convolutional networks. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019*, pages 2145–2150.

Pushkar Mishra, Helen Yannakoudakis, and Ekaterina Shutova. 2019b. Tackling online abuse: A survey of automated abuse detection methods. *arXiv preprint arXiv:1908.06024*.

Edoardo Mosca. 2020. Explainability of hate speech detection models. Master's thesis, Technical University of Munich. Advised and supervised by Maximilian Wich and Georg Groh.

Marzieh Mozafari, Reza Farahbakhsh, and Noël Crespi. 2020. A BERT-Based Transfer Learning Approach for Hate Speech Detection in Online Social Media. *Studies in Computational Intelligence*, 881 SCI:928–940.

Emily R Munro. 2011. The protection of children online: a brief scoping review to identify vulnerable groups. *Childhood Wellbeing Research Centre*.

Manoel Horta Ribeiro, Pedro H Calais, Yuri A Santos, Virgílio AF Almeida, and Wagner Meira Jr. 2018. Characterizing and detecting hateful users on twitter. In *Twelfth international AAAI conference on web and social media*.

Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. 2016. ”Why should I trust you?” Explaining the predictions of any classifier. In *Proc. 22nd ACM SIGKDD Intl. Conf. Knowledge Discovery and Data Mining*, pages 1135–1144.

Julian Risch, Robin Ruff, and Ralf Krestel. 2020. Offensive language detection explained. In *Proceedings of the Second Workshop on Trolling, Aggression and Cyberbullying*, pages 137–143.

Anna Schmidt and Michael Wiegand. 2017. A survey on hate speech detection using natural language processing. In *Proc. 5th Intl. Workshop on Natural Language Processing for Social Media*, pages 1–10.

Lloyd S Shapley. 1953. A value for n-person games. *Contributions to the Theory of Games*, 2(28):307–317.

Ellen Spertus. 1997. Smokey: Automatic recognition of hostile messages. In *Proceedings of Innovative Applications of Artificial Intelligence (IAAI)*, pages 1058–1065.

Erik Štrumbelj and Igor Kononenko. 2014. Explaining prediction models and individual predictions with feature contributions. *Knowledge and information systems*, 41(3):647–665.

Prashanth Vijayaraghavan, Hugo Larochelle, and Deb Roy. 2019. Interpretable Multi-Modal Hate Speech Detection. In *Intl. Conf. Machine Learning AI for Social Good Workshop*.

Cindy Wang. 2018. Interpreting neural network hate speech classifiers. In *Proc. 2nd Workshop on Abusive Language Online*, pages 86–92.

William Warner and Julia Hirschberg. 2012. Detecting hate speech on the world wide web. In *Proceedings of the second workshop on language in social media*, pages 19–26.

Zeerak Waseem. 2016. Are You a Racist or Am I Seeing Things? Annotator Influence on Hate Speech Detection on Twitter. In *Proc. First Workshop on NLP and Computational Social Science*, pages 138–142.

Zeerak Waseem, Thomas Davidson, Dana Warmsley, and Ingmar Weber. 2017. Understanding abuse: A typology of abusive language detection subtasks. *arXiv preprint arXiv:1705.09899*.

Zeerak Waseem and Dirk Hovy. 2016. Hateful symbols or hateful people? predictive features for hate speech detection on twitter. In *Proceedings of the NAACL student research workshop*, pages 88–93.

Maximilian Wich, Jan Bauer, and Georg Groh. 2020. Impact of politically biased data on hate speech classification. In *Proceedings of the Fourth Workshop on Online Abuse and Harms*, pages 54–64.

Matthew L Williams, Pete Burnap, Amir Javed, Han Liu, and Sefa Ozalp. 2020. Hate in the machine: Anti-black and anti-muslim social media posts as predictors of offline racially and religiously aggravated crime. *The British Journal of Criminology*, 60(1):93–117.

A Results on the Davidson Dataset

A.1 Feature Space learned by the Text Model

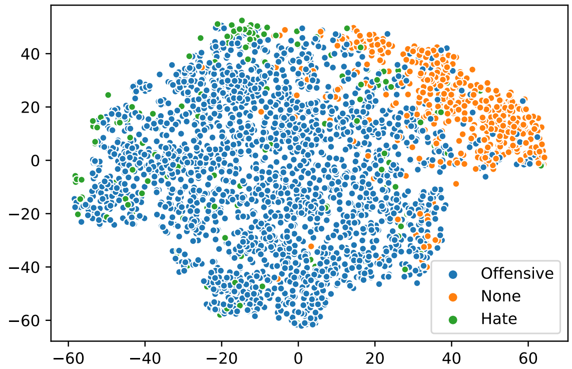


Figure 7: DAVIDSON tweets, colored by label, in the feature space learned by the text model.

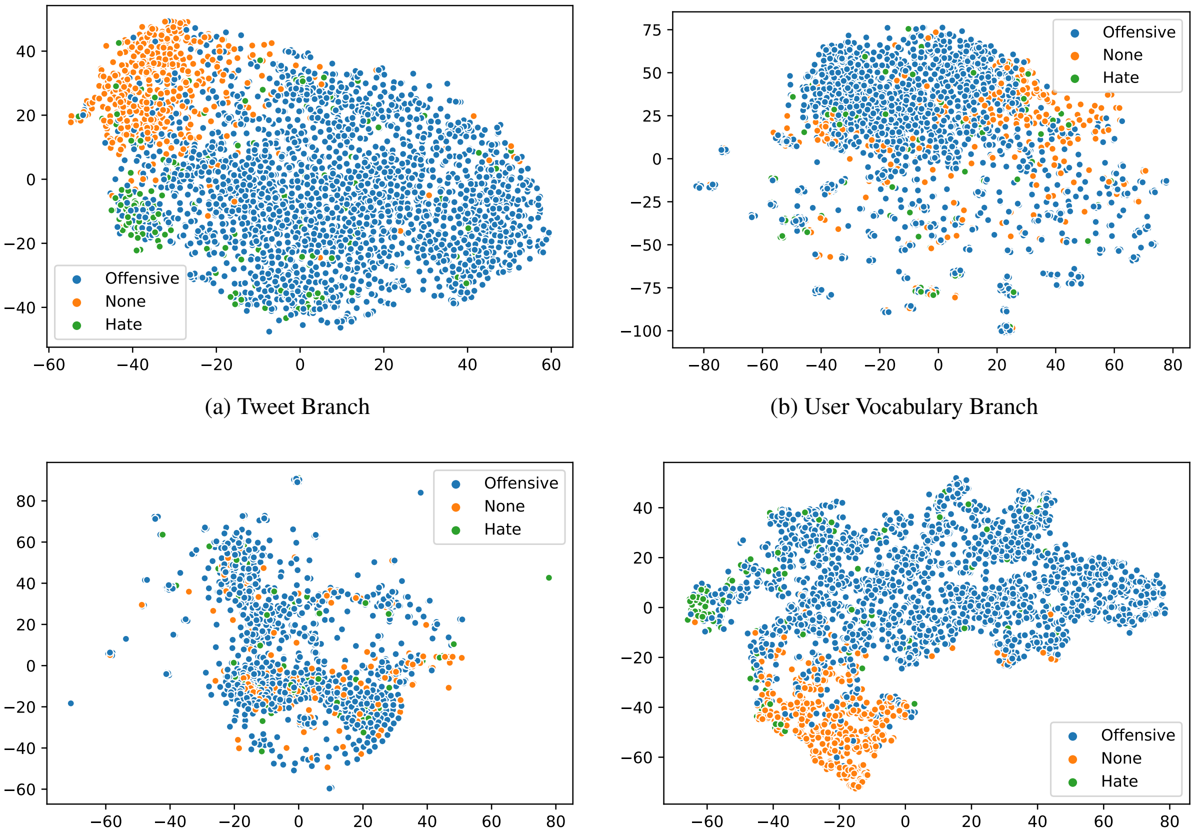
Figure 7 shows the feature space learned by our text model on DAVIDSON. Overall, the distribution looks similar as the one of WASEEM visualized in figure 4d. We can notice that hate tweets are extremely sparse and mixed with the offensive ones. This is reflected by the poor model performance on the hate class, possibly caused by the conceptual overlap that these two classes have. On the other hand, non-harmful tweets are mostly concentrated in one area of the plot, confirming the satisfactory F1 scored achieved.

A.2 Feature Space learned by the Social Model

Figure 8 shows the feature space learned by our social model on DAVIDSON. As done for WASEEM, we report the plots both for the single branches as well as for their combination. The tweet branch (figure 8a) has a similar structure to figure 7. However, hateful tweets are also concentrated in a small portion of the space. This reflects the improved performance that the social model had on the hate class. This suggests that the information coming from the other input sources reinforces the signal backpropagated to the tweet branch, resulting in a less chaotic mixture of hateful and offensive tweets. The user vocabulary (figure 8b) and the follower network branch (figure 8c) do not present the same characteristics as seen on WASEEM. In this case, we do not have the data points separated into multiple clusters. The same goes for the overall learned feature space (figure 8d), where all the tweets are contained in one single cloud. This is consistent with what we observed in terms of F1 Scores. In contrast to what occurred on WASEEM, user features did not cause a substantial impact on the feature space on DAVIDSON and thus did not produce a large leap in performance.

A.3 Complement to Figure 6

Figure 6 compares the model's behavior on the same tweet but with different authors, one racist and one regular. For completeness, figure 9 shows the corresponding plots—Shapley values and embedding onto the features space—for the same tweet when generated by a sexist user. The result is analogous to the one obtained with the regular user. Also in this case the tweet is not classified as racist (12% confidence). The estimated Shapley values show a substantial impact of the user vocabulary against the racism class. The embedding onto the latent space shows once more that changing the author caused the tweet to embed in a different cluster, hence excluding the possibility of the content being classified correctly.



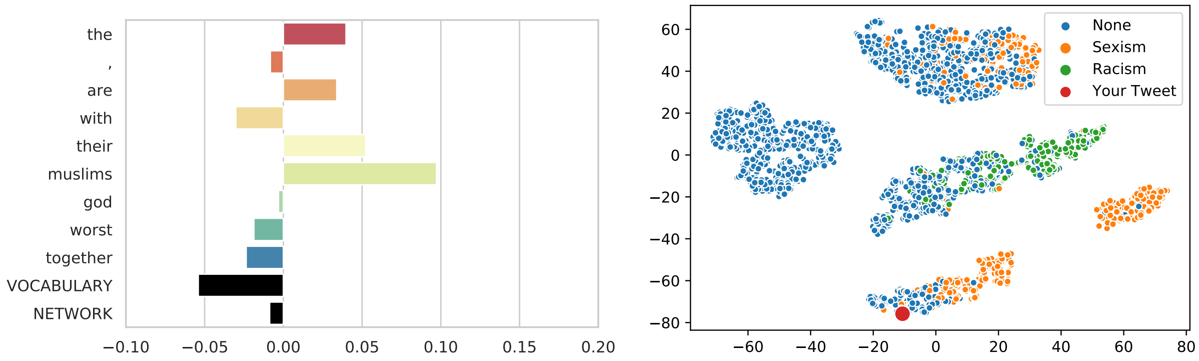
(a) Tweet Branch

(b) User Vocabulary Branch

(c) Follower Network Branch

(d) Last Hidden Layer

Figure 8: Latent space visualization of our social model on DAVIDSON, colored by label. The features are extracted from the single branches before the concatenation: tweet (a), user's vocabulary (b), follower network (c). The last plot (d) shows instead the final learned features space, after all branches are combined and processed together.



(a) Sexist User, Shapley Values

(b) Sexist User, Embedding in Latent Space

Figure 9: Features contribution (w.r.t. racism class) and embeddings of the islamophobic tweet in the social model's latent space. The pair of plots are w.r.t. the prediction done with sexist author.

A.3 STUDY V

Edoardo Mosca, Katharina Hermann, Tobias Eder, and Georg Groh (July 2022). "Explaining Neural NLP Models for the Joint Analysis of Open-and-Closed-Ended Survey Answers." In: *Proceedings of the 2nd Workshop on Trustworthy Natural Language Processing (TrustNLP 2022)*. Seattle, U.S.A.: Association for Computational Linguistics, pp. 49–63. DOI: 10.18653/v1/2022.trustnlp-1.5. URL: https://aclanthology.org/2022.trustnlp-1.5

[3] https://creativecommons.org/licenses/by/4.0/

Publication Summary

"Large-scale surveys are a widely used instrument to collect data from a target audience. Beyond the single individual, an appropriate analysis of the answers can reveal trends and patterns and thus generate new insights and knowledge for researchers. Current analysis practices employ shallow machine learning methods or rely on (biased) human judgment. This work investigates the usage of state-of-the-art NLP models such as BERT to automatically extract information from both open- and closed-ended questions. We also leverage explainability methods at different levels of granularity to further derive knowledge from the analysis model. Experiments on EMS—a survey-based study researching influencing factors affecting a student's career goals—show that the proposed approach can identify such factors both at the input- and higher concept-level." (Mosca, Hermann, et al., 2022, p. 1)

Author Contributions

Edoardo Mosca contributed to the study as follows:

- Conception, development, and lead of the research project **80%**
- Literature review and feasibility study **80%**
- Methodology and experimental design **60%**
- Implementation and interpretation of results. **0%**
- Drafting of the manuscript **80%**
- Submission, peer review, and publication process **100%**

Explaining Neural NLP Models for the Joint Analysis of Open- and Closed-Ended Survey Answers

Edoardo Mosca, Katharina Hermann, Tobias Eder and **Georg Groh**
TU Munich, Department of Informatics, Germany
{edoardo.mosca, katharina.hermann, tobi.eder}@tum.de
grohg@in.tum.de

Abstract

Large-scale surveys are a widely used instrument to collect data from a target audience. Beyond the single individual, an appropriate analysis of the answers can reveal trends and patterns and thus generate new insights and knowledge for researchers. Current analysis practices employ shallow machine learning methods or rely on (biased) human judgment. This work investigates the usage of state-of-the-art NLP models such as BERT to automatically extract information from both open- and closed-ended questions. We also leverage explainability methods at different levels of granularity to further derive knowledge from the analysis model. Experiments on EMS—a survey-based study researching influencing factors affecting a student's career goals—show that the proposed approach can identify such factors both at the input- and higher concept-level.

1 Introduction

Surveys and questionnaires are prevalent tools to inquire about an audience and collect ideas, opinions, and thoughts. Common examples are requesting user feedback concerning a specific product or service, regular reports for scientific studies that involve human subjects, and census questionnaires directed to a certain demographic population.

Carrying out an appropriate and thorough analysis of the collected answers is of major relevance for researchers both in the industry and academia. However, the generated data are often a combination of open-ended and closed-ended questions. While the former gathers a participant's thoughts in text form, the latter consists in selecting one (or more) of the options specified by the survey designer. Utilizing both types remains a popular choice as closed-ended questions are very suitable to derive statistical conclusions but may lack details which are in turn provided by open-ended answers.

Currently, the two dominant analysis practices comprise traditional closed-vocabulary and open-vocabulary methods (Eichstaedt et al., 2021). Whereas the former introduces human biases and is resource-intensive, the latter overcomes these challenges with the help of *Natural Language Processing* (NLP) techniques. Nonetheless, both approaches fail to consider contextual information and do not leverage currently available NLP architectures to deal with more complex patterns.

In this work, we bridge the gap in research and investigate the usage of deep-learning-based methods from NLP and explainability techniques to extract knowledge and interpret correlations from surveys presenting both structured and unstructured components. Our contribution can be summarized as follows:

(1) We apply a popular transformer architecture (DistilBERT) (Sanh et al., 2019) to open-ended questions. This enables our approach to extract contextual correlations from the text with high precision compared to traditional methods.

(2) Due to the model's black-box characteristics, we utilize post-hoc explainability methods to interpret the extracted correlations. Specifically, we utilize several variants of *SHapley Additive exPlanations* (SHAP) (Lundberg and Lee, 2017) to analyze both instance-level feature importance as well as high-level concepts learned by the model (Yeh et al., 2020). These methods are applied to several components to generate a holistic understanding of the model used for the analysis.

(3) Our approach delivers promising results on the EMS 1.0 dataset - studying influencing factors in students' career goals (Gilmartin et al., 2017). First, it identifies the most relevant factors from closed-ended responses with high precision. Second, it also automatically reveals influencing factors from the open-ended text answers.

Proceedings of the 2nd Workshop on Trustworthy Natural Language Processing (TrustNLP 2022), pages 49 - 63
July 14, 2022 ©2022 Association for Computational Linguistics

2 Related Work

2.1 The EMS Study and Entrepreneurial Behavior Predictors

In this paper, we work with the *Engineering Major Survey* (EMS) longitudinal study of students' career goals by Gilmartin et al. (2017). Analysis of the contents of this study was previously conducted mainly by the social sciences with a focus on qualitative approaches to extract the most influential variables on career goals (Grau et al., 2016; Levine et al., 2017). Quantitative correlation between variables was previously explored by Atwood et al. (2020) relating *Social Cognitive Career Theory* (SCCT) (Lent et al., 1994) to different predefined topics for the purpose of survey design, such as students demographics, first-generation status, and family background. Schar et al. (2017) meanwhile focused on the variables *Engineering Task Self-Efficacy* and *Innovation Self-Efficacy* through explainable regression models.

2.2 Analysis of Open-ended Survey Question in the Social Sciences

In the social sciences, textual analysis has a long history of utilizing manual analysis methods such as *Grounded Theory Method* (GMT) Bryant and Charmaz (2007). However recently, automated text analysis has been used for both open- and closed-vocabulary methods.

Closed-vocabulary methods: Analysis is done by working with a hand-crafted closed-vocabulary such as LIWC (Pennebaker et al., 2001) and calculating the relative frequencies of dictionaries with respect to the text (Eichstaedt et al., 2021).

Open-vocabulary methods: Following the GMT method, these approaches aim to discover topics from data, rather than from a predefined word list (Roberts et al., 2014). For instance, Guetterman et al. (2018) uses NLP techniques such as topic modeling and clustering for textual analysis of survey questions. These approaches were mostly utilizing well-known bag-of-words methods such as *Latent Dirichlet Allocation* (LDA) (Blei et al., 2003) and *Latent Semantic Analysis* (LSA) (Deerwester et al., 1990). Further work included clustering semantic distances in adjectives for situation-taxonomies (Parrigon et al., 2017).

2.3 Post-Hoc Explainability

Methods from *eXplainable Artificial Intelligence* (XAI) (Arrieta et al., 2020; Mosca et al., 2021) have recently gained popularity as deep architectures—such as transformers—behave like black-boxes (Brown et al., 2020; Devlin et al., 2019). In particular, post-hoc explainability techniques are able to explain the *why* behind a certain prediction even if the model is not inherently interpretable.

The literature has classified existing interpretability approaches in structured taxonomies depending on their core characteristics (Madsen et al., 2021; Doshi-Velez and Kim, 2017). We identify the following two broad categories as the most relevant for our research objectives and methodology.

Feature attribution methods: They assign each input feature with a relevance score describing its importance for the model prediction. Approaches such as SAGE (Covert et al., 2020) and GAM (Ibrahim et al., 2019) produce global explanations, i.e. at the dataset level. Others, instead, focus on generating insights at the instance-level, i.e. about a specific model prediction. Prominent local methods are LIME (Ribeiro et al., 2016) and SHAP (Lundberg and Lee, 2017).

Concept-based methods: Concept-oriented techniques aim at extracting human-interpretable concepts, consisting of sets of (text) features from several input samples sharing similar activation patterns within the model. Prominent approaches are TCAV (Kim et al., 2018), ACE (Ghorbani et al., 2019), and ConceptSHAP (Yeh et al., 2020). The latter is unsupervised—i.e. it does not require a predefined list of concepts to test for—and thus particularly relevant for our methodology.

Please note that these explainability techniques can be applied to the whole model—i.e. from input to output—or sub-components of it, such as (groups of) layers and neurons (Sajjad et al., 2021).

3 Methodology

3.1 EMS Data

We use the EMS 1.0 data as our data source and prediction target. The EMS study 1.0 from 2015 consists of data from 7,197 students enrolled across 27 universities in the United States. The study poses a mix of closed and free-text questions across 8 different topics, ranging from background characteristics to self-efficacy and career goals. More de-

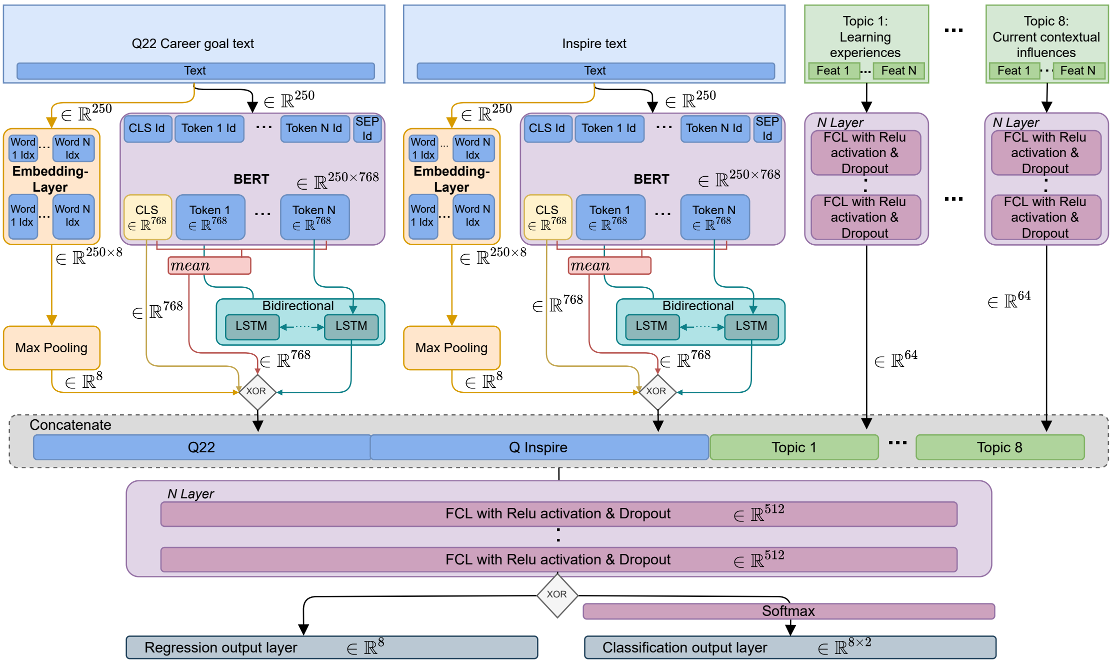


Figure 1: Model architecture combining both text and numerical (i.e. categorical) feature classification architectures. The XORs indicate different model choices for various sub-components.

tailed descriptions of these questions can be found in Gilmartin et al. (2017) or in a more condensed form in Appendix A of this paper.

While most of the questions in the survey are multiple-choice, referred to as *numerical* or *categorical*, two questions require open-text answers. *Q22* asks about the short-term plans of students within five years of graduating while the *Inspire* question, asks how the survey itself influenced the thought process of the students towards their career goals.

The independent variable we are trying to predict is *Q20* also named *Career goal* in the survey and asks for the likelihood of a person to pursue a career in 8 distinct circumstances, ranging from corporate employee to non-profit founder. Each of these cases is given a Likert score from 0 to 4 representing the likelihood from *highly unlikely* to *very likely*. In our model, we use both the numerical responses from the 8 topics as well as the free-text answers to predict career preferences.

3.2 Model Architecture

The architecture for the prediction task is illustrated in Figure 1 and can be split into three logical parts. The first section (top left) deals with the open text variables and is based on DistilBERT and embedding layers. The second input section (top right), processes the numerical features pertinent to each topic through a series of *Fully Connected* (FC) layers.

After being processed in parallel, the latent representations of each open-text question and each topic are concatenated and processed through another FC block, before generating the final prediction.

The output is generated by two distinct heads: a regression task trained on mean absolute error loss approximating the numerical values of the subquestions of *Q20* and a classification output trained with a cross-entropy loss, predicting general favorable or unfavorable tendencies. In each case, there are eight individual outputs for each prediction, one for each task.

Open-end text variables: The main part of the text processing architecture is based on DistilBERT (Sanh et al., 2019), which is utilized without fine-tuning to create text representations for the following layers. The four branching architecture choices in this part include **(1)** the use of the embedding vector encoding the CLS token, **(2)** mean averaging over word token embedding vectors (Wolf et al., 2020), **(3)** feeding the word token vectors through a BiLSTM layer (Graves and Schmidhuber, 2005) and **(4)** a single eight-dimensional embedding layer trained on the free-text task data.

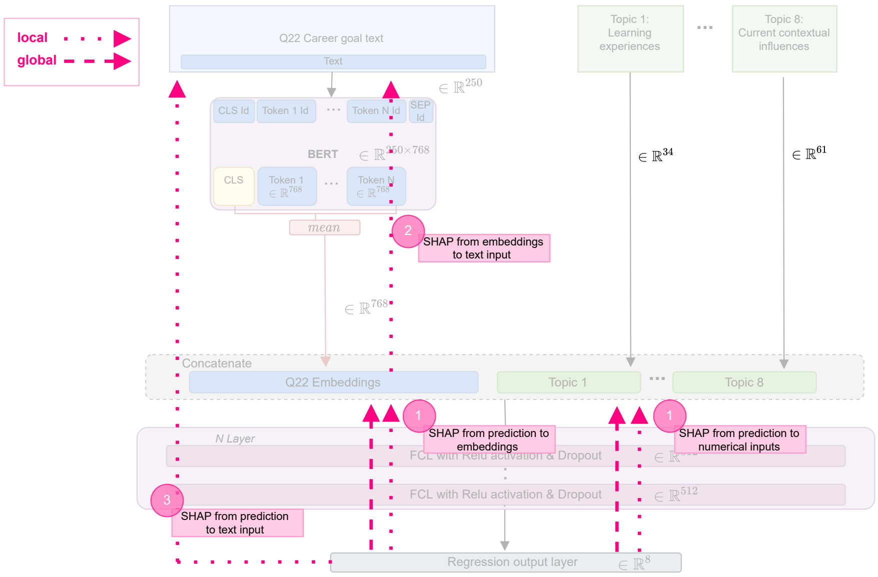

local
global
Q22 Career goal text
Text
$\in \mathbb{R}^{250}$
Topic 1: Learning experiences
Topic 8: Current contextual influences
CLS Id
Token 1 Id
Token N Id
SEP Id
BERT
$\in \mathbb{R}^{250\times768}$
CLS
Token 1 $\in \mathbb{R}^{768}$
Token N $\in \mathbb{R}^{768}$
$mean$
$\in \mathbb{R}^{34}$
$\in \mathbb{R}^{61}$
2
SHAP from embeddings to text input
$\in \mathbb{R}^{768}$
Concatenate
Q22 Embeddings
Topic 1
Topic 8
1
SHAP from prediction to embeddings
1
SHAP from prediction to numerical inputs
N Layer
FCL with Relu activation & Dropout
$\in \mathbb{R}^{512}$
FCL with Relu activation & Dropout
$\in \mathbb{R}^{512}$
3
SHAP from prediction to text input
Regression output layer
$\in \mathbb{R}^{8}$

Figure 2: Explainability experiments with SHAP values for different parts of the model. **(1)** Global and local SHAP values from prediction to intermediate layer with embeddings and numerical features as inputs, **(2)** local SHAP values from embeddings to text input, **(3)** local SHAP values from prediction to text input

Numerical feature variables: This part of the architecture takes all recorded numerical features (minus the covariate) as input and groups them by topic according to the SCCT framework. Each topic is fed through separate FC layer model streams before being concatenated with the representation from the text variables. While most features can be input directly as a single value, some represent nominal choices and are input as one-hot encoding vectors instead.

3.3 Model Explanations

We apply several post-hoc explainability methods to both explain specific model predictions and gain a holistic understanding of what our model has learned.

Low-level feature and neuron explanations We employ SHAP (Lundberg and Lee, 2017) to compute local and global feature relevance explanations. This enables us to quantify the most important input components in terms of overall model accuracy, but also to identify the features dominating a specific prediction (Wich et al., 2021). Specifically, we **(1)** calculate and compare SHAP values for both the text and numerical value embeddings. Then, we **(2)** look at which parts of the text input trigger the neurons presenting the highest activation in the previous analysis. Finally, we **(3)** compute SHAP values for the input text w.r.t. the final model prediction. Figure 2 shows a detailed overview of all SHAP explanation experiments and how they relate to the various model inputs and inner components.

High-level concept explanations: We utilize ConceptSHAP (Yeh et al., 2020) to understand how the model captures and organizes higher-level information for its predictions. This information is extracted in the form of concepts, i.e. clusters of embedding vectors each summarized by a concept vector c_i which acts as the cluster's centroid. Beyond their extraction, we **(1)** use the K nearest neighbors of c_i to describe each concept, **(2)** measure the influence of each concept for a single prediction, and **(3)** report *completeness scores* - i.e. how well the set of extracted concepts describe the model's behavior (Yeh et al., 2020). Analogous to Figure 2 for SHAP experiments, Figure 12 (See Appendix C) shows a detailed overview of all ConceptSHAP explanation experiments and how they relate to the various model inputs and inner components.

4 Results

Results are presented in two distinct sections. Firstly, we present the numerical results for the prediction task in the case of both the regression and the classification heads for the whole architecture. The performance here is evaluated through

Architecture			T1	T2	T3	T4	T5	T6	T7	T8
Q22	no T	C	51.66	60.10	56.89	44.61	48.40	51.85	52.50	63.70
		R	53.82	51.36	50.82	58.75	43.63	42.24	46.71	62.40
Ins.	no T	C	46.66	38.20	40.68	42.20	50.21	43.48	46.08	42.69
		R	42.26	39.79	36.07	37.77	37.10	41.79	41.88	35.48
Q22+Ins.	no T	C	45.69	59.87	52.31	53.11	47.92	59.71	50.91	51.12
		R	**63.48**	47.46	50.59	45.20	41.06	41.29	39.86	58.73
No text	all T	C	50.85	53.34	61.03	52.40	57.03	**67.88**	61.02	72.65
		R	50.79	54.17	61.58	57.33	**58.94**	56.91	59.08	74.65
Q22	all T	C	63.01	60.74	**63.53**	**60.87**	50.77	57.76	54.90	73.64
		R	59.69	**63.64**	59.59	55.84	56.62	56.03	**62.66**	**76.23**
Ins.	all T	C	57.23	59.08	57.63	54.22	54.68	57.48	65.30	69.24
		R	48.33	47.00	51.49	50.45	48.92	46.12	58.49	72.47
Q22+Ins.	all T	C	58.71	57.52	59.86	55.51	55.16	58.56	62.40	71.55
		R	59.49	54.62	63.27	55.50	56.83	49.58	56.60	73.61

Table 1: F1 Scores for the combined model, utilizing different parts of the input data. Architectures differ based on which parts of the input they use. Question 22 (Q22) and Question Inspire (Ins.) are free text questions, tabular data (T) is counted separate. All numbers are reported for performance on classification (C) and regression (R) tasks. Best model for each task (T1 to T8) in bold.

macro F1 score for all eight individual topic predictions. Secondly, we show explanations for these model predictions through explainability frameworks SHAP and ConceptSHAP.

4.1 Task Performance

We conducted a variety of experiments on different sub-parts of the architecture and finally on different overall combinations of features for the architecture presented in Figure 1.

Text-based prediction We tested four different configurations of the free-text part of the model architecture, each with a different mode to generate embeddings as described in section 3.2. Results are taken individually for each of the eight tasks and for both regression and classification heads. A stripped-down version of these results for task 8 *Founding for-profit* can be found in Table 2. The full table of results can be found in Appendix D.

	CLS	Mean	BiLSTM	Embedding
C	60.66	**63.70**	37.88	49.66
R	53.96	62.40	58.18	50.27

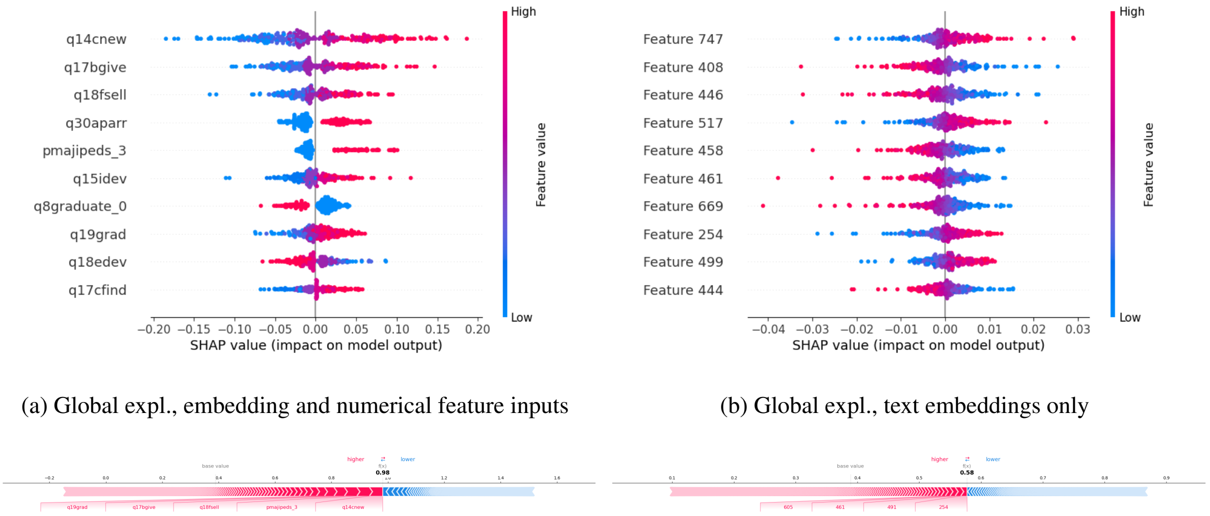
Table 2: F1 Scores for the Q22 text input, predicting task 8 (T8) for each architecture. Best model in bold.

In summary, the mean average model performed best on the label 8 task, scoring an F1 score of 63.70% for the classification and 62.40% for the regression task. On six of the other tasks, the *mean*-model performed better than the other models. The classification task was overall easier to achieve, yielding higher scores across all tasks with the notable exception of task 4.

Numerical variable-based prediction In this part of the evaluation, we ran the numerical variable part of the architecture without any text inputs to compare results on the 8 tasks (T1 to T8). We evaluated the input of each of the 8 SCCT topics individually, as well as on the combination of all topics for prediction.

The best performing model utilized all available topics concatenated directly before processing with a mean F1 score of 72.65% (C) for the classification and 74.65% (R) for the regression head on task 8. The full list of results is available in Appendix D. Based on the numerical variables only, it is unclear whether the classification or the regression head performed better overall since performance turned out to be highly task and architecture-dependent.

Combined performance The overall performance of the model is evaluated for a variety of feature combinations. For all the cases we chose the best performing combinations of the architecture for text-based prediction and the concatenated input of all SCCT topics for the numerical variable input. The combination of possible features is then for text input either *no text*, *Q22*, the *Inspire* ques-



(a) Global expl., embedding and numerical feature inputs

(b) Global expl., text embeddings only

(c) Local explanation: all features

(d) Local explanation: text embeddings only

Figure 3: SHAP values for all features (left) and text embedding only (right). Global explanations (top) and local explanations (bottom). The higher in magnitude the value is, the more important a feature is for the model, while a positive value contributes to a prediction value of 1 and a negative value to a class value of 0. See appendix E for a larger scale version of (c) and (d).

tion, as well as all numerical topic variables or none of them resulting in 8 total possible combinations.

The full evaluation of these input variations is shown in Table 1. Best results are achieved by the model combining *Q22* text input with the full set of SCCT topics, resulting in a macro F1 score of 73.64% (C) for classification and 76.23% (R) for regression. The *Inspire* text variable instead contributes negatively across tasks as well as scoring the worst for singular performance at 42.69% (C) and 35.48% (R) F1 score. Our best model thus uses all available numerical features, as well as the free-text input from *Q22* as input, processing the DistilBERT embedding into a mean sentence embedding vector and a regression head output for prediction.

4.2 Interpretability examples

For simplicity, we present explanations for the model reporting the best performance (see Table 1). For the first set of feature attribution explanations, we focus on the eighth head—capturing the *likelihood of starting a for-profit company*. For the concept-based explanations, instead, we examine all heads as concepts describe the information captured by the model overall.

Low-level feature and neuron explanations We begin by looking at the global importance of numerical features and text embeddings w.r.t. the model prediction. As one can see in Figure 3, the ten most important features are numerical features and no single embedded word is as relevant for the model. This is coherent with the observation in section 4.1 that additionally considering text led only to a slight performance improvement. Moreover, we can observe that the four most relevant features are *q14new*, *q17give*, *q18sell*, and *q30aparr*, which are particularly related with entrepreneurial behavior.

Figure 3 also shows two local explanations resulting from the first experiment. These again show the SHAP values for the text embeddings and the numerical features. The colors indicate whether the features push the prediction in a positive (pink for class 1) or negative (blue for class 0) direction. The strength of each feature's contribution is indicated by the length of its corresponding segment. Taking variable *q14cnew* as an example, low feature values impact the model negatively, while high values impact it positively, while in-between feature values land in between those values.

Examples of local explanations generated by the second and third experiments are visualized in Figure 4. In particular, we can observe the text features' influence both on the most influential neuron identified in the first experiments (4a) and on the

Concept	Nearest neighbors	Word cloud
1	want to be successful. find a job my own business no thanks work hard ill do whatever. no concrete plans yet run my own business. no comments no idea	software (5), my (6), no (17), thanks (6), idea (5), company (5), have (6), work (7)
2	i want to attend medical school i plan to find a mechanical i am planning to be a product i plan on working as a i would like to go into manufacturing and continue education with goal i would first like to pursue doctoral degree having my own company i will be starting a career as an seeking law degree, to move into	I (63), my (13), work (10), plan (24), find (5), graduate (8), will (17), be (17), go (7), am (5), career (6), get (6), job (7), would (13), like (14), engineering (7), working (13)
3	business learn skills, turn hobbies into i hope to run my own business start a company overseas earn experience in a small .. either go into industry or go gain experience in the industry. would like to get into management own company when i have the expertise my feet in a start up company early a good paying job at a company that	company (19), my (13), industry (14), work (22), engineering (18), start (12), I (21), business (6), go (12), own (6), job (9), pursue (5), will (8), plan (6), engineer (5), get (7), degree (6), masters (5), working (13), be (5)
4	school within the next two years. work there for 3 years in the next five years i hope work abroad at some point. 5 to 6 years. at least the next two years, i there for at least three years. tentative at that point in time i want in the next five years i field at least once.	at (19), my (13), go (12), industry (14), work (22), engineering (18), start (12), I (21), business (6), engineer (5), be (5), own (6), job (9), pursue (5), will (8), plan (6), get (7), degree (6), masters (5), working (13)

Table 3: The four concepts with 10 examples from the top 100 nearest neighbors and the word clouds containing the most frequent words from the nearest neighbors

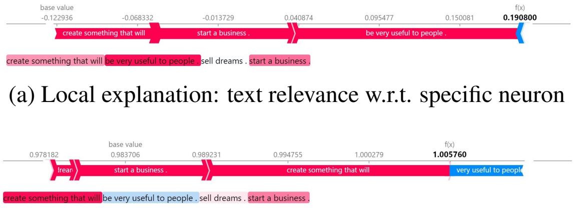
model's output (4b). It is instructive to notice that—in contrast to the model as a whole—SHAP values w.r.t. to this specific neuron are all non-negative. This indicates that this unit has specialized in capturing only positive features, i.e. desire to start a for-profit company.

Higher-level concept explanations While ConceptSHAP (Yeh et al., 2020) does not require a predefined list of concepts, we still need to manually set how many we want to model. We choose four as we are seeking to extract broad and general concepts.

For each concept, we look at the 100 nearest neighbors' word embeddings. We then map these back to their corresponding word token and include four neighboring tokens from their corresponding sentence. Furthermore, we count the word tokens appearing in the top 100 nearest neighbors and construct a word cloud with the ones occurring more than five times.

Once the concepts have been extracted automatically, they can be inspected manually by humans who can look for a common theme in the word cloud and the nearest neighbors. Table 3 presents an overview of the extracted concepts via showing the ten nearest neighbors in addition to the word cloud extracted from the top 100.

The first concept mainly contains nearest neighbors describing a lack of orientation and concrete career plans. Indeed, "no" is one of the words dominating this word cloud. The second, in contrast, captures a strong sense of having a clear path for the own future career. Here, most sentences start



(a) Local explanation: text relevance w.r.t. specific neuron

(b) Local explanation: text relevance w.r.t. model output

Figure 4: Local SHAP values describing the impact of the embedding layer and numerical feature inputs on the model's prediction for 4 different samples, 2 belonging to class 0 (not wanting to start a for-profit company) and 2 belonging to class 1 (wanting to start a for-profitcompany). See E for a larger scale version.

with "I" and contain words like "will" and "plan", indicating strong traits of self-centeredness and determination. Both these concepts match what also discovered by Grau et al. (2016, p.8): i.e. the *clarity of plans*.

The third concept revolves around the plan type rather than its certainty or concreteness. For instance, we find general words like "company", "work", and "engineering", which indicate the goal of founding a company, joining a startup, or working in the industry. This matches the idea of *career characteristics*, also found in Grau et al. (2016, p.8). Finally, the last concept is the most distinctive as it captures the *plan timeline*, clearly present in all the nearest neighbors listed. This concept, connecting career plans to the time dimension, cannot be found in previous works such as Grau et al. (2016). The completeness scores achieved by these concepts are reported in the appendix (see C).

5 Discussion and Comparison

We employed several architectures to solve the the problem of career choice prediction to improve over prevailing closed and open-vocabulary methods. While for some survey responses correlations were strenuous, we found general success in predicting variables relating to entrepreneurial aspirations.

We see an overall increase in performance by combining textual and numerical input data. While numerical data is generally more predictive in our experiments, the 119 numerical variables are also a lot more nuanced than the free-text answers *Q22* and *Inspire*. Despite this, prediction from text alone still manages to perform relatively well across different tasks. The negative impact on performance of including the *Inspire* variable in models is likely due to the limited amount of text in the answers to the question.

To back up our model findings with explanations, we applied SHAP and ConceptSHAP as post-hoc approaches. The first confirmed what we observed in terms of model performance and provided us with a good understanding of the global and local relevance of each component: numerical features, text features, and embeddings. The second, instead, led to the identification of relevant concepts —*clarity of plans*, *career characteristics*, and *plan timeline*—in line with the human judgment of previous works.

6 Conclusion and Future Work

This work investigated the usage of state-of-the-art NLP and XAI techniques for analyzing user-generated survey data. Instead of manually examining individual answers, our methodology heavily relies on analyzing and interpreting a predictor model trained to extract correlations and patterns from the whole data set. We proposed a multi-modal architecture consisting of a DistilBERT transformer architecture and FC layers. The former is used to extract information from open-ended textual answers while the latter process the numerical features representing closed-ended answers. The model achieves satisfactory accuracy in predicting students' career goals and aspirations.

We leveraged SHAP and ConceptSHAP to generate both instance-level and concept-level explanations. These methods were applied at different levels of granularity to assemble a holistic understanding of the model's reasoning. Experiments on the EMS survey show promising results in predicting the students' entrepreneurial ambition. Moreover, local explanations provide us insights about the most relevant questions overall as well as relevant factors w.r.t. a single student. The automatic high-level concept analysis also led to insightful findings which were very similar to what was found in previous research including human judgment.

We release our code to the public to facilitate further research and development [1].

Acknowledgments

This paper is based on a joint work in the context of Katharina Hermann's master's thesis (Hermann, 2022).

[1] https://github.com/EdoardoMosca/explainable-ML-survey-analysis

References

Alejandro Barredo Arrieta, Natalia Díaz-Rodríguez, Javier Del Ser, Adrien Bennetot, Siham Tabik, Alberto Barbado, Salvador García, Sergio Gil-López, Daniel Molina, Richard Benjamins, et al. 2020. Explainable artificial intelligence (xai): Concepts, taxonomies, opportunities and challenges toward responsible ai. *Information Fusion*, 58:82–115.

Sara A Atwood, Shannon K Gilmartin, Angela Harris, and Sheri Sheppard. 2020. Defining first-generation and low-income students in engineering: An exploration. In *ASEE Annual Conference proceedings*.

David M Blei, Andrew Y Ng, and Michael I Jordan. 2003. Latent dirichlet allocation. *Journal of machine Learning research*, 3(Jan):993–1022.

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

Antony Bryant and Kathy Charmaz. 2007. *The Sage handbook of grounded theory*. Sage.

Ian Covert, Scott M Lundberg, and Su-In Lee. 2020. Understanding global feature contributions with additive importance measures. *Advances in Neural Information Processing Systems*, 33:17212–17223.

Scott Deerwester, Susan T Dumais, George W Furnas, Thomas K Landauer, and Richard Harshman. 1990. Indexing by latent semantic analysis. *Journal of the American society for information science*, 41(6):391–407.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *NAACL-HLT (1)*.

Finale Doshi-Velez and Been Kim. 2017. Towards a rigorous science of interpretable machine learning. *arXiv preprint arXiv:1702.08608*.

Johannes C Eichstaedt, Margaret L Kern, David B Yaden, HA Schwartz, Salvatore Giorgi, Gregory Park, Courtney A Hagan, Victoria A Tobolsky, Laura K Smith, Anneke Buffone, et al. 2021. Closed-and open-vocabulary approaches to text analysis: A review, quantitative comparison, and recommendations. *Psychological Methods*, 26(4):398.

Amirata Ghorbani, James Wexler, James Y Zou, and Been Kim. 2019. Towards automatic concept-based explanations. *Advances in Neural Information Processing Systems*, 32.

Shannon K Gilmartin, Helen L Chen, Mark F Schar, Qu Jin, George Toye, A Harris, Emily Cao, Emanuel Costache, Maximillian Reithmann, and Sheri D Sheppard. 2017. Designing a longitudinal study of engineering students' innovation and engineering interests and plans: The engineering majors survey project. ems 1.0 and 2.0 technical report. *Stanford University Designing Education Lab, Stanford, CA, Technical Report*.

Michelle Marie Grau, Sheri Sheppard, Shannon Katherine Gilmartin, and Beth Rieken. 2016. What do you want to do with your life? insights into how engineering students think about their future career plans. In *2016 ASEE Annual Conference & Exposition*.

Alex Graves and Jürgen Schmidhuber. 2005. Framewise phoneme classification with bidirectional lstm and other neural network architectures. *Neural networks*, 18(5-6):602–610.

Timothy C Guetterman, Tammy Chang, Melissa DeJonckheere, Tanmay Basu, Elizabeth Scruggs, and VG Vinod Vydiswaran. 2018. Augmenting qualitative text analysis with natural language processing: methodological study. *Journal of medical Internet research*, 20(6):e9702.

Katharina Hermann. 2022. Explaining neural nlp models to understand students' career choices. Master's thesis, Technical University of Munich. Advised and supervised by Edoardo Mosca and Georg Groh.

Mark Ibrahim, Melissa Louie, Ceena Modarres, and John Paisley. 2019. Global explanations of neural networks: Mapping the landscape of predictions. In *Proceedings of the 2019 AAAI/ACM Conference on AI, Ethics, and Society*, pages 279–287.

Been Kim, Martin Wattenberg, Justin Gilmer, Carrie Cai, James Wexler, Fernanda Viegas, et al. 2018. Interpretability beyond feature attribution: Quantitative testing with concept activation vectors (tcav). In *International conference on machine learning*, pages 2668–2677. PMLR.

Robert W Lent, Steven D Brown, and Gail Hackett. 1994. Toward a unifying social cognitive theory of career and academic interest, choice, and performance. *Journal of vocational behavior*, 45(1):79–122.

Amber Levine, T Bjorklund, Shannon Gilmartin, and Sheri Sheppard. 2017. A preliminary exploration of the role of surveys in student reflection and behavior. In *Proceedings of the American Society for Engineering Education Annual Conference, June 25-28. Columbus, OH.*

Scott M. Lundberg and Su-In Lee. 2017. A unified approach to interpreting model predictions. *NeurIPS 2017.*

Andreas Madsen, Siva Reddy, and Sarath Chandar. 2021. Post-hoc interpretability for neural nlp: A survey. *arXiv preprint arXiv:2108.04840*.

Edoardo Mosca, Maximilian Wich, and Georg Groh. 2021. Understanding and interpreting the impact of user context in hate speech detection. In *Proceedings of the Ninth International Workshop on Natural Language Processing for Social Media*, pages 91–102.

Scott Parrigon, Sang Eun Woo, Louis Tay, and Tong Wang. 2017. Caption-ing the situation: A lexically-derived taxonomy of psychological situation characteristics. *Journal of personality and social psychology*, 112(4):642.

James W Pennebaker, Martha E Francis, and Roger J Booth. 2001. Linguistic inquiry and word count: Liwc 2001. *Mahway: Lawrence Erlbaum Associates*, 71(2001):2001.

Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. 2016. " why should i trust you?" explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, pages 1135–1144.

Margaret E Roberts, Brandon M Stewart, Dustin Tingley, Christopher Lucas, Jetson Leder-Luis, Shana Kushner Gadarian, Bethany Albertson, and David G Rand. 2014. Structural topic models for open-ended survey responses. *American journal of political science*, 58(4):1064–1082.

Hassan Sajjad, Narine Kokhlikyan, Fahim Dalvi, and Nadir Durrani. 2021. Fine-grained interpretation and causation analysis in deep nlp models. *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies: Tutorials*.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. *NeurIPS 2017*.

Mark Schar, S Gilmartin, Beth Rieken, S Brunhaver, H Chen, and Sheri Sheppard. 2017. The making of an innovative engineer: Academic and life experiences that shape engineering task and innovation self-efficacy. In *Proceedings of the American Society for Engineering Education Annual Conference, June 25-28. Columbus, OH.*

Maximilian Wich, Edoardo Mosca, Adrian Gorniak, Johannes Hingerl, and Georg Groh. 2021. Explainable abusive language classification leveraging user and network data. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 481–496. Springer.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, et al. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 conference on empirical methods in natural language processing: system demonstrations*, pages 38–45.

Chih-Kuan Yeh, Been Kim, Sercan Arik, Chun-Liang Li, Tomas Pfister, and Pradeep Ravikumar. 2020. On completeness-aware concept-based explanations in deep neural networks. *Advances in Neural Information Processing Systems*, 33:20554–20565.

A Appendix: Details on the EMS 1.0 survey data

The longitudinal *Engineering Major Survey* (EMS) by Gilmartin et al. (2017) consists of three surveys in total, conducted between 2015 and 2019. In this paper we only focus on the EMS 1.0 data from 2015 consisting of 7197 surveyed students of engineering enrolled at 27 universities in the US. The study is based on the *Social Cognitive Career Theory* (SCCT) framework (Lent et al., 1994) about how a students decision making is influenced by 8 specific topics.

These topics are:

- Topic 1: Learning experiences
- Topic 2: Self-efficacy (Engineering task, professional/interpersonal, innovation)
- Topic 3: Innovation outcome expectations
- Topic 4: Background characteristics / influences (gender, ethnicity, family background)
- Topic 5: Innovation interests
- Topic 6: Career Goals: Innovative work
- Topic 7: Job Targets
- Topic 8: Current contextual influences (major, institutional, peer)

Independent variables: Our independent variables come from topic 7 and surmise the following question *Q20*: "How likely is it that you will do each of the following in the first five years after you graduate?". It provides eight career possibilities which constitute our tasks 1 through 8 for each of the prediction heads:

1. Work as an employee for a small business or start-up company.

2. Work as an employee for a medium- or large-size business.

3. Work as an employee for a non-profit organization (excluding a school or college/university).

4. Work as an employee for the government, military, or public agency (excluding a school or college/university).

5. Work as a teacher or educational professional in a K-12 school.

6. Work as a faculty member or educational professional in a college or university.

7. Found or start your own for-profit organization.

8. Found or start your own non-profit organization.

Each entry can be answered with a Likert scale score ranging from 0 *'Definitely will not'* to 4 *'Definitely will'*.

For classification, the 5 classes (0 through 4) are binned into a binary label: low interest and high interest. The binning is done depending on the median of each label as illustrated in Figure 5. However this strategy ultimately still leads to unbalanced classes in some cases.

Lastly, we also analyze Pearson Correlation between all remaining labels after list-wise deletion, to determine whether they can be considered unique tasks. Our analysis illustrated in Figure 6 illustrated this point with most classes showing low correlation (less than 0.5).

Numerical variables: There are 119 numerical feature variables that operate on a categorical or five-point scale split across 30 distinct questions. Scale design, as well as the order of questions was based on minimizing bias in survey response.

An additional test of correlation between numerical features and task labels showed only weak linear correlation, indicating that solving the task is more complex.

Open text variables: We consider two open text variables, which are the following:

1. *Q22*: "We have asked a number of questions about your future plans. If you would like to elaborate on what you are planning to do, in the next five years or beyond, please do so here."

2. *Inspire*: "To what extent did this survey inspire you to think about your education in new or different ways? Please describe."

While these questions nominally fall under topic 7 in the SCCT framework, we treat them as disjoint topics during processing.

We additionally evaluated text length and correlation between the description of tasks of our target variable and the contents of the free text fields. Text length does not correlate with our label classes as shown in Figure 7. At the same time we could detect some correlation through keyword matching with *Q22*, especially relating to a lower score. Meanwhile there is no strong correlation between keywords for the Inspire variable. Results of the correlation analysis can be found in Figure 8 and Figure 9.

B Appendix: Non-combined architectures

This appendix shows the schematics for both architectures which omit either the textual or numerical variable part which was used for the detailed experiments listed in Appendix D. The text-only architecture can be found in Figure 10 while the numerical-only model can be found in Figure 10.

C Appendix: Higher-Level ConceptSHAP Experiments

Figure 12 shows an overview of the experiments involving ConceptSHAP (Yeh et al., 2020). Completeness scores for the retrieved concepts are reported in Table 4.

D Appendix: Detailed experiment results

This section lists the full results for the text-only classification and regression tasks across topics in table 5 as well as the results for the numerical variable prediction in table 6.

E Further SHAP Examples

To improve their readability, we now present again the SHAP force plots already included in 4.2. We also present further examples not previously included.

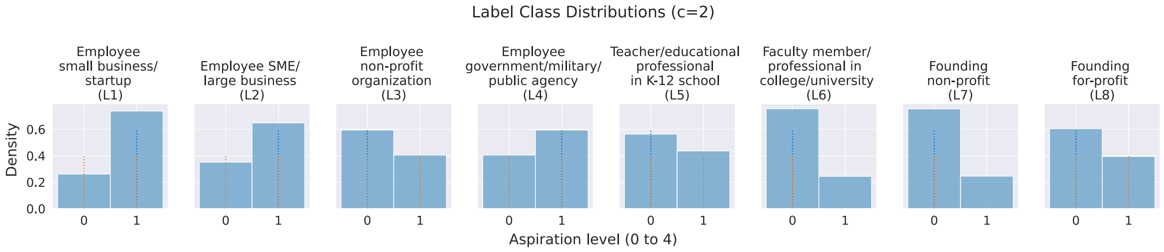


Figure 5: Splits binning 5 classes into two by median for each task.

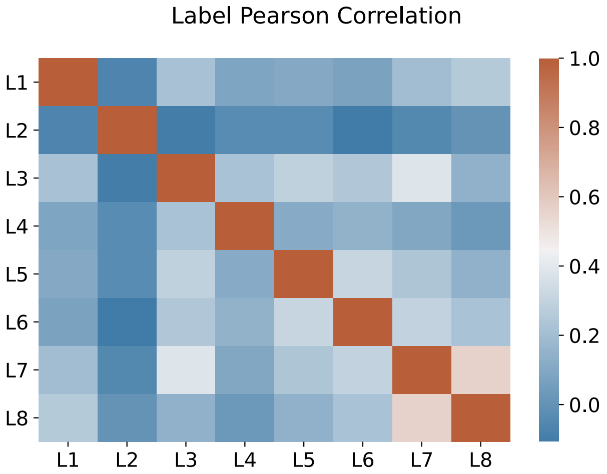


Figure 6: Pearson Correlation between each of the 8 labels. Values range from 0.0 to 1.0.

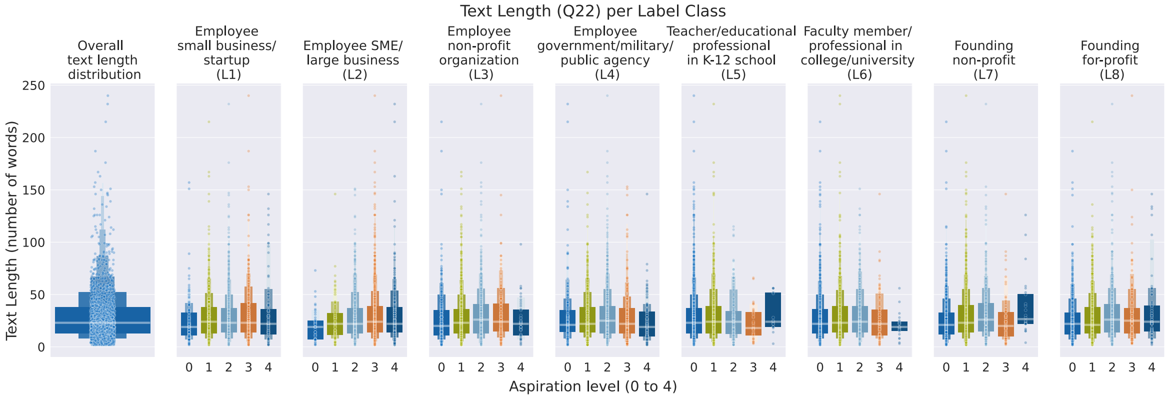


Figure 7: Overall text length distribution of Q22 and distribution grouped by classes per label.

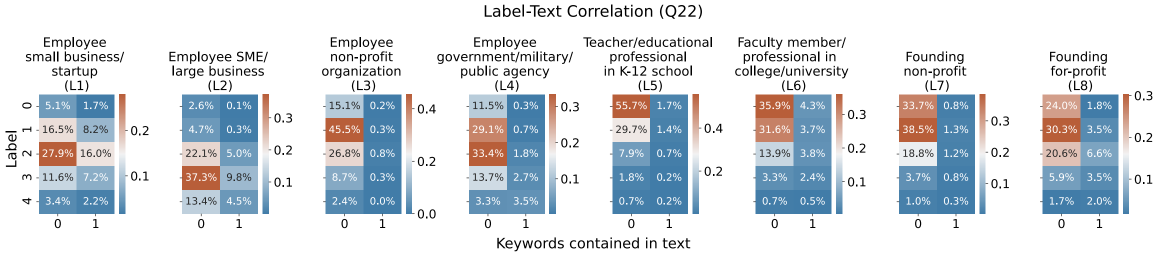


Figure 8: Model architecture for numerical features with FC layers.

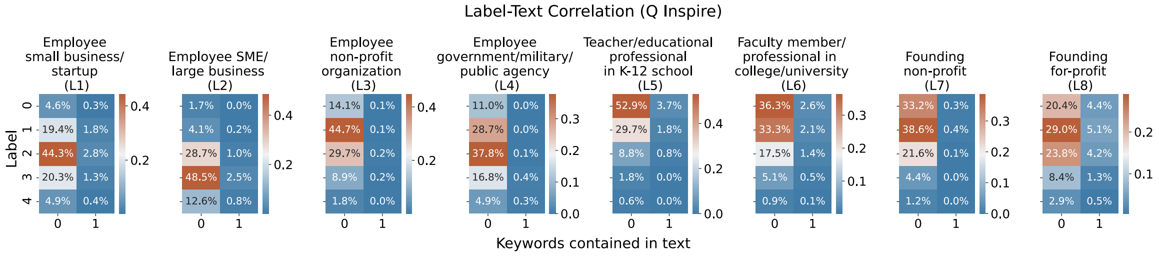


Figure 9: Model architecture for numerical features with FC layers.

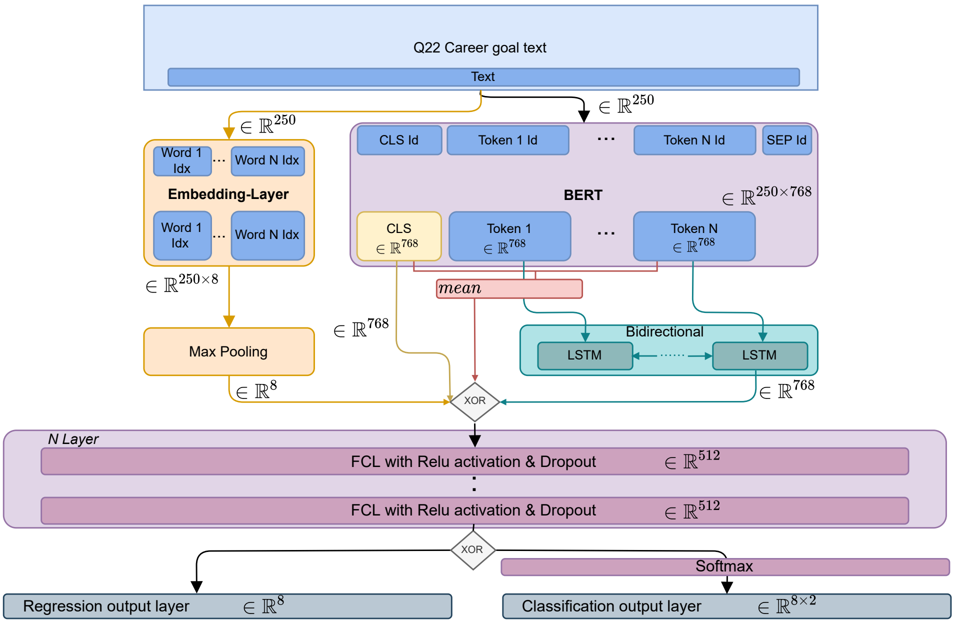

Q22 Career goal text
Text
$\in \mathbb{R}^{250}$
$\in \mathbb{R}^{250}$
Word 1 Idx
Word N Idx
Embedding-Layer
Word 1 Idx
Word N Idx
$\in \mathbb{R}^{250\times8}$
Max Pooling
$\in \mathbb{R}^{8}$
CLS Id
Token 1 Id
Token N Id
SEP Id
BERT
$\in \mathbb{R}^{250\times768}$
CLS $\in \mathbb{R}^{768}$
Token 1 $\in \mathbb{R}^{768}$
Token N $\in \mathbb{R}^{768}$
mean
$\in \mathbb{R}^{768}$
Bidirectional
LSTM
LSTM
$\in \mathbb{R}^{768}$
XOR
N Layer
FCL with Relu activation & Dropout $\in \mathbb{R}^{512}$
FCL with Relu activation & Dropout $\in \mathbb{R}^{512}$
XOR
Softmax
Regression output layer $\in \mathbb{R}^{8}$
Classification output layer $\in \mathbb{R}^{8\times2}$

Figure 10: Model architecture for prediction through text processing. The XOR signifies different model choices w.r.t. different embedding processing steps and different output heads.

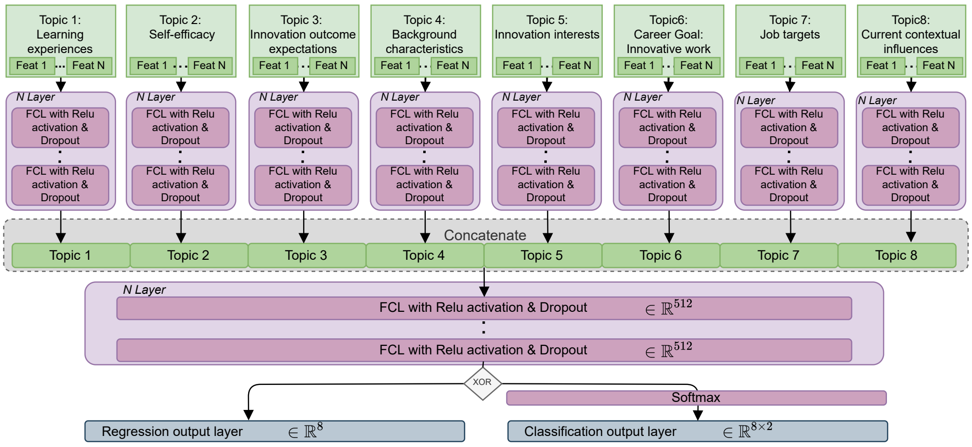

Topic 1: Learning experiences
Topic 2: Self-efficacy
Topic 3: Innovation outcome expectations
Topic 4: Background characteristics
Topic 5: Innovation interests
Topic6: Career Goal: Innovative work
Topic 7: Job targets
Topic8: Current contextual influences
Feat 1
Feat N
N Layer
FCL with Relu activation & Dropout
FCL with Relu activation & Dropout
Concatenate
Topic 1
Topic 2
Topic 3
Topic 4
Topic 5
Topic 6
Topic 7
Topic 8
N Layer
FCL with Relu activation & Dropout $\in \mathbb{R}^{512}$
FCL with Relu activation & Dropout $\in \mathbb{R}^{512}$
XOR
Softmax
Regression output layer $\in \mathbb{R}^{8}$
Classification output layer $\in \mathbb{R}^{8\times2}$

Figure 11: Model architecture for numerical features with FC layers. The XOR indicates the different model choices w.r.t. different output heads choices.

L1	L2	L3	L4	L5	L6	L7	L8
-0.66	-0.79	0.17	-0.59	0.18	0.93	0.89	0.73

Table 4: The completeness scores for each of the 8 prediction heads measuring how well the concepts can be used to recover predictions from the original model **(3)**

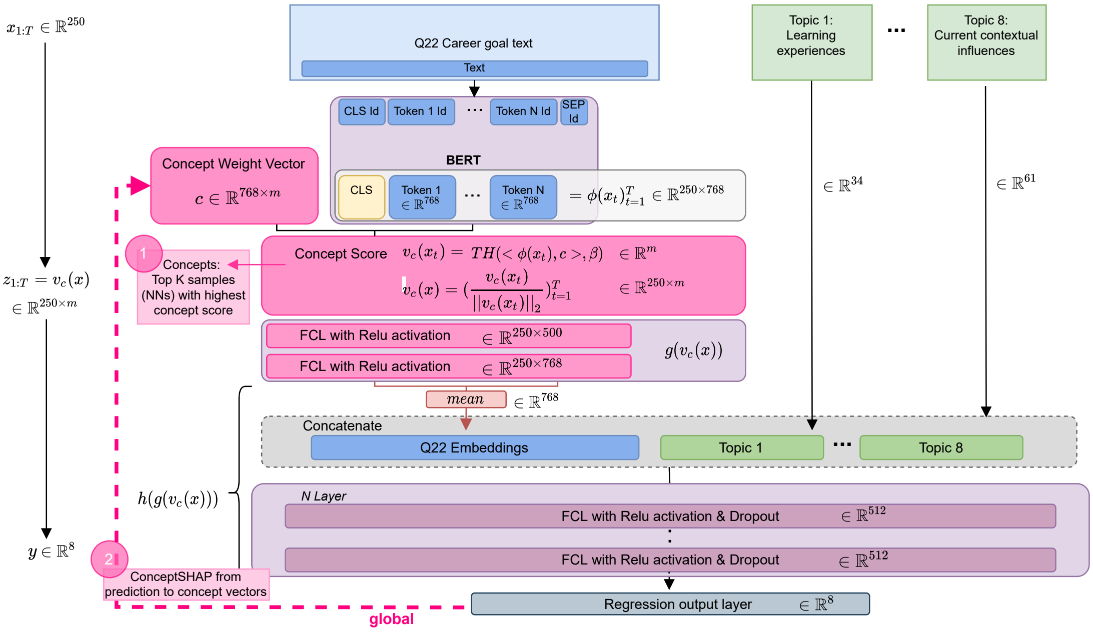
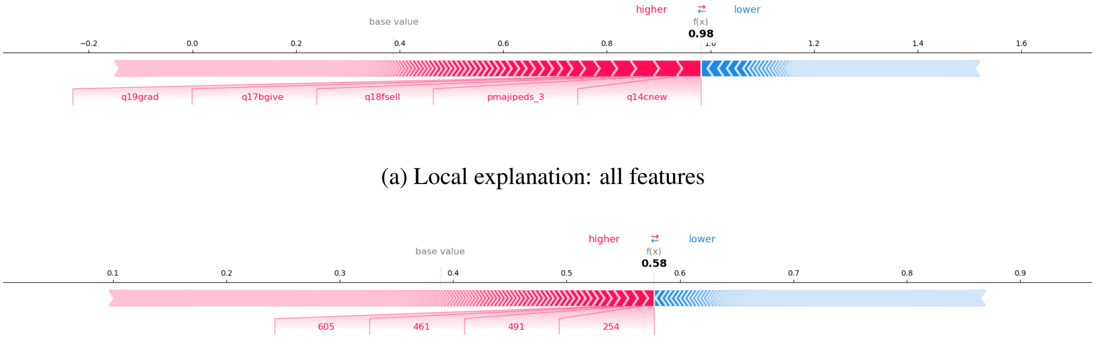

$x_{1:T} \in \mathbb{R}^{250}$
Q22 Career goal text
Text
Topic 1: Learning experiences
...
Topic 8: Current contextual influences
CLS Id
Token 1 Id
...
Token N Id
SEP Id
BERT
Concept Weight Vector
$c \in \mathbb{R}^{768 \times m}$
CLS
Token 1 $\in \mathbb{R}^{768}$
Token N $\in \mathbb{R}^{768}$
$= \phi(x_t)_{t=1}^T \in \mathbb{R}^{250 \times 768}$
$\in \mathbb{R}^{34}$
$\in \mathbb{R}^{61}$
1
Concepts: Top K samples (NNs) with highest concept score
Concept Score $v_c(x_t) = TH(<\phi(x_t), c>, \beta) \in \mathbb{R}^m$
$v_c(x) = (\frac{v_c(x_t)}{||v_c(x_t)||_2})_{t=1}^T \in \mathbb{R}^{250 \times m}$
$z_{1:T} = v_c(x)$
$\in \mathbb{R}^{250 \times m}$
FCL with Relu activation $\in \mathbb{R}^{250 \times 500}$
FCL with Relu activation $\in \mathbb{R}^{250 \times 768}$
$g(v_c(x))$
$mean$ $\in \mathbb{R}^{768}$
Concatenate
Q22 Embeddings
Topic 1
...
Topic 8
$h(g(v_c(x)))$
N Layer
FCL with Relu activation & Dropout $\in \mathbb{R}^{512}$
FCL with Relu activation & Dropout $\in \mathbb{R}^{512}$
$y \in \mathbb{R}^8$
2
ConceptSHAP from prediction to concept vectors
Regression output layer $\in \mathbb{R}^8$
global

Figure 12: Explainability experiments with a concept-based method called ConceptSHAP. The original model is extended to a surrogate model to train concept vectors c_j , which function as the centroids of the concepts. These concepts are then being formed by the top k nearest neighbour tokens embeddings to the concept vectors **(1)**. In addition to the pure concept extraction, we can measure their importance for the prediction of the model by using the principle of SHAP, **(2)**.

		T1	T2	T3	T4	T5	T6	T7	T8
CLS	C	57.12	58.05	48.49	48.17	42.26	46.42	44.74	60.66
	R	54.05	51.26	36.41	44.24	35.21	42.74	43.44	53.96
mean	C	51.66	**60.10**	**56.89**	44.61	**48.40**	**51.85**	**52.50**	**63.70**
	R	53.82	51.36	50.82	**58.75**	43.63	42.24	46.71	62.40
BiLSTM	C	42.75	38.74	39.17	37.73	35.36	43.11	42.18	37.88
	R	52.82	54.49	36.70	49.77	34.91	42.38	42.62	58.18
embedding	C	**54.57**	47.62	50.52	50.06	48.31	48.05	46.45	49.66
	R	52.21	47.68	47.83	43.04	48.06	43.56	51.22	50.27

Table 5: F1 Scores for the Q22 text input, predicting all tasks. Best model for each task in bold.



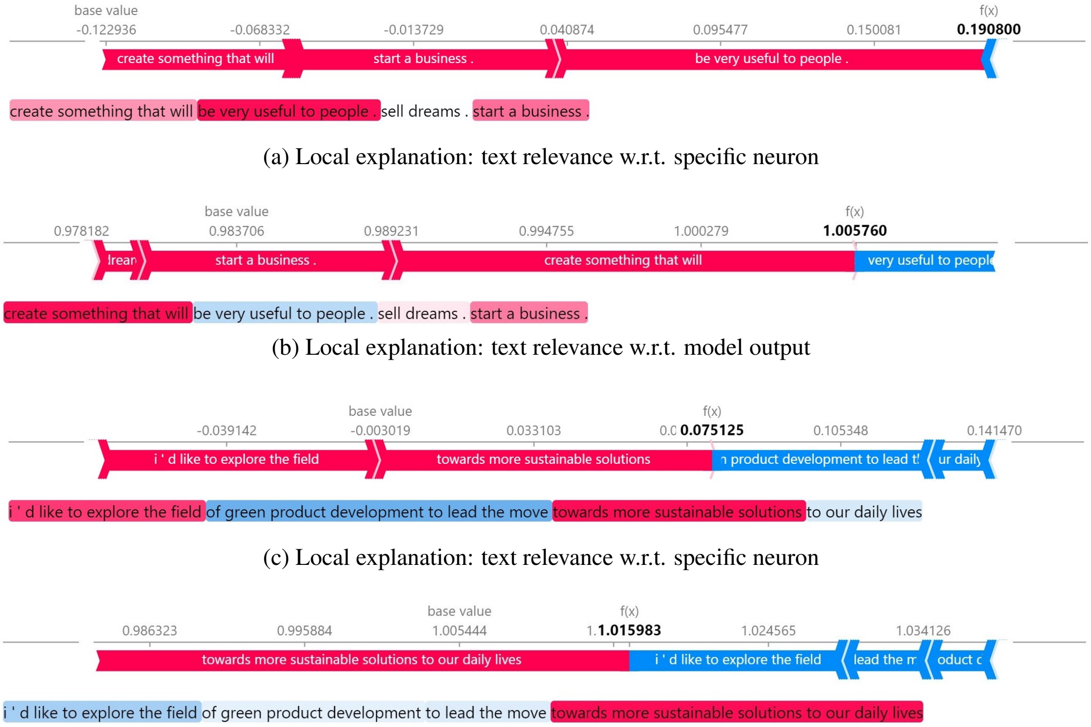
(a) Local explanation: all features

(b) Local explanation: text embeddings only

Figure 13: Larger scale version of plots (c) and (d) from Figure 3

		T1	T2	T3	T4	T5	T6	T7	T8
topic 1	C	44.39	41.74	57.46	41.36	52.21	49.62	58.07	66.83
	R	41.63	40.48	44.78	42.77	44.04	44.92	46.51	64.92
topic 2	C	48.42	44.83	54.36	42.51	40.32	42.60	42.74	62.39
	R	43.56	39.98	36.46	38.16	35.17	43.68	43.09	55.41
topic 3	C	42.74	46.80	48.03	42.17	54.85	46.05	48.10	50.18
	R	43.33	39.68	45.84	38.02	48.54	46.42	47.09	48.60
topic 4	C	42.28	39.33	45.18	47.16	45.22	44.94	44.71	54.37
	R	42.17	40.39	44.03	51.85	41.54	48.91	48.24	48.06
topic 5	C	44.94	51.00	58.68	45.98	55.64	46.77	43.44	64.33
	R	44.33	48.75	53.58	41.33	51.86	43.06	43.51	62.98
topic 6	C	49.26	40.35	44.68	47.18	38.39	42.71	42.57	57.70
	R	44.36	44.39	37.53	38.89	35.85	42.46	43.16	61.96
topic 7	C	46.40	61.60	56.66	46.20	54.11	58.03	43.02	44.29
	R	47.32	**62.69**	50.31	51.28	52.65	60.86	43.59	48.98
topic 8	C	46.41	44.39	52.06	51.84	45.58	45.68	44.04	48.69
	R	48.92	56.72	49.96	53.97	49.65	53.29	43.37	38.91
all topics sep.	C	51.41	60.80	60.90	**57.35**	**61.06**	60.79	59.29	70.25
	R	**51.81**	55.66	52.38	56.31	52.84	55.83	53.32	67.74
dir.	C	50.85	53.34	61.03	52.40	57.03	**67.88**	**61.02**	72.65
	R	50.79	54.17	**61.58**	57.33	58.94	56.92	59.08	**74.65**

Table 6: F1 Scores for the numerical data differing on inputs only. Best model for each task in bold.



(a) Local explanation: text relevance w.r.t. specific neuron

(b) Local explanation: text relevance w.r.t. model output

(c) Local explanation: text relevance w.r.t. specific neuron

(d) Local explanation: text relevance w.r.t. model output

Figure 14: Larger scale version of SHAP plots presented in Figure 4. Two additional examples have also been added - i.e. (c) and (d).

A.4 STUDY VII

Edoardo Mosca, Shreyash Agarwal, Javier Rando Ramırez, and Georg Groh (May 2022). "“That Is a Suspicious Reaction!”: Interpreting Logits Variation to Detect NLP Adversarial Attacks." In: *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Dublin, Ireland: Association for Computational Linguistics, pp. 7806–7816. DOI: 10.18653/v1/2022.acl-long.538. URL: https://aclanthology.org/2022.acl-long.538

[4] https://creativecommons.org/licenses/by/4.0/

Publication Summary

"Adversarial attacks are a major challenge faced by current machine learning research. These purposely crafted inputs fool even the most advanced models, precluding their deployment in safety-critical applications. Extensive research in computer vision has been carried to develop reliable defense strategies. However, the same issue remains less explored in natural language processing. Our work presents a model-agnostic detector of adversarial text examples. The approach identifies patterns in the logits of the target classifier when perturbing the input text. The proposed detector improves the current state-of-the-art performance in recognizing adversarial inputs and exhibits strong generalization capabilities across different NLP models, datasets, and word-level attacks." (Mosca, Agarwal, et al., 2022, p. 1)

Author Contributions

Edoardo Mosca contributed to the study as follows:

- Conception, development, and lead of the research project **100%**
- Literature review and feasibility study **80%**
- Setup and implementation of experiments **20%**
- Analysis and interpretation of results. **30%**
- Drafting of the manuscript **70%**
- Submission, peer review, and publication process **100%**

"That Is a Suspicious Reaction!": Interpreting Logits Variation to Detect NLP Adversarial Attacks

Edoardo Mosca
TU Munich,
Department of Informatics,
Germany
edoardo.mosca@tum.de

Shreyash Agarwal
TU Munich,
Department of Informatics,
Germany
shreyash.agarwal@tum.de

Javier Rando-Ramirez
ETH Zurich,
Department of Computer Science,
Switzerland
jrando@student.ethz.ch

Georg Groh
TU Munich,
Department of Informatics,
Germany
grohg@in.tum.de

Abstract

Adversarial attacks are a major challenge faced by current machine learning research. These purposely crafted inputs fool even the most advanced models, precluding their deployment in safety-critical applications. Extensive research in computer vision has been carried to develop reliable defense strategies. However, the same issue remains less explored in natural language processing. Our work presents a model-agnostic detector of adversarial text examples. The approach identifies patterns in the logits of the target classifier when perturbing the input text. The proposed detector improves the current state-of-the-art performance in recognizing adversarial inputs and exhibits strong generalization capabilities across different NLP models, datasets, and word-level attacks.

1 Introduction

Despite recent advancements in *Natural Language Processing* (NLP), adversarial text attacks continue to be highly effective at fooling models into making incorrect predictions (Ren et al., 2019; Wang et al., 2019; Garg and Ramakrishnan, 2020). In particular, syntactically and grammatically consistent attacks are a major challenge for current research as they do not alter the semantical information and are not detectable via spell checkers (Wang et al., 2019). While some defense techniques addressing this issue can be found in the literature (Mozes et al., 2021; Zhou et al., 2019; Wang et al., 2019), results are still limited in performance and text attacks keep evolving. This naturally raises concerns around the safe and ethical deployment of NLP systems in real-world processes.

Previous research showed that analyzing the model's logits leads to promising results in discriminating manipulated inputs (Wang et al., 2021; Aigrain and Detyniecki, 2019; Hendrycks and Gimpel, 2016). However, logits-based adversarial detectors have been only studied on computer vision applications. Our work transfers this type of methodology to the NLP domain and its contribution can be summarized as follows:

(1) We introduce a logits-based metric called *Word-level Differential Reaction* (WDR) capturing words with a suspiciously high impact on the classifier. The metric is model-agnostic and also independent from the number of output classes.

(2) Based on WDR scores, we train an adversarial detector that is able to distinguish original from adversarial input texts preserving syntactical correctness. The approach substantially outperforms the current state of the art in NLP.

(3) We show our detector to have full transferability capabilities and to generalize across multiple datasets, attacks, and target models without needing to retrain. Our test configurations include transformers and both contextual and genetic attacks.

(4) By applying a post-hoc explainability method, we further validate our initial hypothesis—i.e. the detector identifies patterns in the WDR scores. Furthermore, only a few of such scores carry strong signals for adversarial detection.

Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics
Volume 1: Long Papers, pages 7806 - 7816
May 22-27, 2022 ©2022 Association for Computational Linguistics

2 Background and Related Work

2.1 Adversarial Text Attacks

Given an input sample x and a target model f, an adversarial example $x' = x + \Delta x$ is generated by adding a perturbation Δx to x such that $\arg\max f(x) = y \neq y' = \arg\max f(x')$. Although this is not required by definition, in practice the perturbation Δx is often imperceptible to humans and x' is misclassified with high confidence. In the NLP field, Δx consists in adding, removing, or replacing a set of words or characters in the original text. Unlike image attacks—vastly studied in the literature (Zhang et al., 2020) and operating in high-dimensional continuous input spaces—text perturbations need to be applied on a discrete input space. Therefore, gradient methods used for images such as FGSM (Goodfellow et al., 2014) or BIM (Kurakin et al., 2017) are not useful since they require a continuous space to perturb x. Based on the text perturbation introduced, text attacks can be distinguished into two broad categories.

Visual similarity: These NLP attacks generate adversarial samples x' that look similar to their corresponding original x. These perturbations usually create typos by introducing perturbations at the character level. DeepWordBug (Gao et al., 2018), HotFlip (Ebrahimi et al., 2018) , and VIPER (Eger et al., 2019) are well-known techniques belonging to this category.

Semantic similarity: Attacks within this category create adversarial samples by designing sentences that are semantically coherent to the original input and also preserve syntactical correctness. Typical word-level perturbations are deletion, insertion, and replacement by synonyms (Ren et al., 2019) or paraphrases (Iyyer et al., 2018). Two main types of adversarial search have been proposed. *Greedy algorithms* try each potential replacement until there is a change in the prediction (Li et al., 2020; Ren et al., 2019; Jin et al., 2020). On the other hand, *genetic algorithms* such as Alzantot et al. (2018) and Wang et al. (2019) attempt to find the best replacements inspired by natural selection principles.

2.2 Defense against Adversarial Attacks in NLP

Defenses based on spell and syntax checkers are successful against character-level text attacks (Pruthi et al., 2019; Wang et al., 2019; Alshemali and Kalita, 2019). In contrast, these solutions are not effective against word-level attacks preserving language correctness (Wang et al., 2019). We identify methods against word-level attacks belonging to two broad categories:

Robustness enhancement: The targeted model is equipped with further processing steps to not be fooled by adversarial samples without identifying explicitly which samples are adversarial. For instance, *Adversarial Training* (AT) (Goodfellow et al., 2014) consists in training the target model also on manipulated inputs. The *Synonym Encoding Method* (SEM) (Wang et al., 2019) introduces an encoder step before the target model's input layer and trains it to eliminate potential perturbations. Instead, *Dirichlet Neighborhood Ensemble* (DNE) (Zhou et al., 2020) and *Adversarial Sparse Convex Combination* (ASCC) (Dong et al., 2021) augment the training data by leveraging the convex hull spanned by a word and its synonyms.

Adversarial detection: Attacks are explicitly recognized to alert the model and its developers. Adversarial detectors were first explored on image inputs via identifying patterns in their corresponding Shapley values (Fidel et al., 2020), activation of specific neurons (Tao et al., 2018), and saliency maps (Ye et al., 2020). For text data, popular examples are *Frequency-Guided Word Substitution* (FGWS) (Mozes et al., 2021) and *learning to DIScriminate Perturbation* (DISP) (Zhou et al., 2019). The former exploits frequency properties of replaced words, while the latter uses a discriminator to find suspicious tokens and uses a contextual embedding estimator to restore the original word.

2.3 Logits-Based Adversarial Detectors

Inspecting output logits has already led to promising results in discriminating between original and adversarial images (Hendrycks and Gimpel, 2016; Pang et al., 2018; Kannan et al., 2018; Roth et al., 2019). For instance, Wang et al. (2021) trains a recurrent neural network that captures the difference in the logits distribution of manipulated samples. Aigrain and Detyniecki (2019), instead, achieves good detection performance by feeding a simple three-layer neural network directly with the logit activations.

Our work adopts a similar methodology but focuses instead on the NLP domain and thus text attacks. In this case (1) logits-based metrics to identify adversarial samples should be tailored to

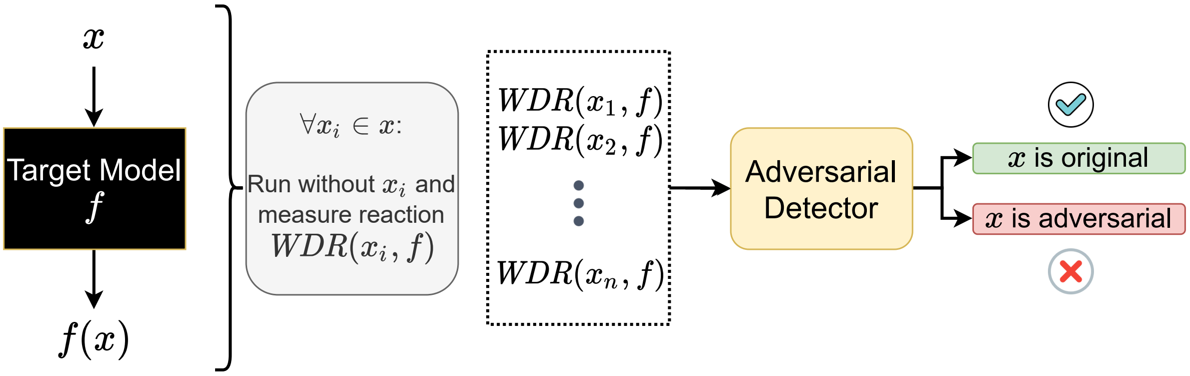


Figure 1: Overview of the proposed method.

the new type of input and (2) detectors should be tested on currently used NLP models such as transformers (Devlin et al., 2019).

3 Methodology

The defense approach proposed in this work belongs to the category of *adversarial detection*. It defends the target model from attacks generated via word-level perturbations belonging to the *semantic similarity* category. The intuition behind the method is that the model's reaction to original- and adversarial samples is going to differ even if the inputs are similar. Hence, it relies on *feature attribution explanations* coupled with a machine learning model to learn such difference and thus identify artificially crafted inputs.

Figure 1 shows the overall pipeline of the approach. Given a text classifier f trained on the task at hand, the pipeline's goal is to detect whether the currently fed input x is adversarial. In 3.1, we explain in greater detail how we measure the model f's reaction to a given input x. This quantity—later indicated with $WDR(x, f)$—is then passed to the adversarial detector, whose training procedure is described in 3.2. Finally, in 3.3, we provide detailed information about the setup of our experiments such as target models, datasets, and attacks.

3.1 Interpreting the Target Model and Measuring its Reaction: Word-Level Differential Reaction

Adversarial attacks based on semantic similarity replace the smallest number of words possible to change the target model's prediction (Alzantot et al., 2018). Thus, we expect the replacements transforming x into x' to play a big role for the output. If not, we would not have $f(x')$ substantially different from $f(x)$. To assess the reaction of the target model f to a given input x, we measure the impact of a word via the *Word-level Differential Reaction* (WDR) metric. Specifically, the effect of replacing a word x_i on the prediction

$$y^* = \arg\max_y p(y|x)$$

is quantified by

$$WDR(x_i, f) = f(x \backslash x_i)_{y^*} - \max_{y \neq y^*} f(x \backslash x_i)_y$$

where $f(x \backslash x_i)_y$ indicates the output logit for class y for the input sample x without the word x_i. Specifically, x_i is replaced by an *unknown word token*. If x is adversarial, we could expect to find perturbed words to have a negative $WDR(x_i, f)$ as without them the input text should recover its original prediction. Table 1 shows an example pair of original and adversarial text together with their corresponding $WDR(x_i, f)$ scores. The original class is recovered after removing a perturbed word in the adversarial sentence. This switch results in a negative WDR. However, even if the most important word is removed from the original sentence (*'worst'*), the predicted class does not change and thus $WDR(x_i, f) > 0$.

Our adversarial detector takes as input $WDR(x, f)$, i.e. the sorted list of WDR scores $WDR(x_i, f)$ for all words x_i in the input sentence. As sentences vary in length, we pad the list with zeros to ensure a consistent input length for the detector.

3.2 Adversarial Detector Training

The adversarial detector is a machine-learning classifier that takes the model's reaction $WDR(x, f)$ as input and outputs whether the input x is adversarial or not. To train the model, we adopt the following multi-step procedure:

Original sentence: Neg. Review (*Class 0*)

This is absolutely the worst trash I have ever seen. It took 15 full minutes before I realized that what I was seeing was a sick joke! [...]

Removed Word x_i	**Logit *Class 0***	**Logit *Class 1***	**WDR $WDR(x_i, f)$**
$\emptyset$	3.44	-3.46	**6.89**
worst	1.68	-1.75	**3.43**
sick	3.34	-3.42	**6.76**
absolutely	3.40	-3.45	**6.86**
realized	3.41	-3.47	**6.89**

Adversarial sentence: Pos. Review (*Class 1*)

This is absolutely the tough trash I have ever seen. It took 15 full minutes before I realized that what I was seeing was a silly joke! [...]

Removed Word x_i	**Logit *Class 0***	**Logit *Class 1***	**WDR $WDR(x_i, f)$**
$\emptyset$	-1.85	2.17	**4.02**
tough	2.14	-1.50	**-3.64**
silly	1.38	-1.37	**-2.75**
absolutely	-0.31	0.48	**0.79**
realized	-1.07	1.36	**2.43**

Table 1: $WDR(x_i, f)$ scores computed for an original sentence and its corresponding adversarial perturbation. Results show how when removing adversarial words such as *tough* or *silly*, the original class is recovered and the WDR becomes negative. $\emptyset$ corresponds to the prediction without any replacements

(S1) Generation of adversarial samples: Given a target classifier f, for each original sample available x, we generate one adversarial example x'. This leads to a balanced dataset containing both normal and perturbed samples. The labels used are *original* and *adversarial* respectively.

(S2) WDR computation: For each element of the mixed dataset, we compute the $WDR(x, f)$ scores as defined in Section 3.1. Once more, this step creates a balanced dataset containing the WDR scores for both normal and adversarial samples.

(S3) Detector training: The output of the second step **(S2)** is split into training and test data. Then, the training data is fed to the detector for training along with the labels defined in step **(S1)**.

Please note that no assumption on f is made. At the same time, the input of the adversarial detector—i.e. the WDR scores—does not depend on the number of output classes of the task at hand. Hence, the adversarial detector is model-agnostic w.r.t. the classification task and the classifier targeted by the attacks.

In our case, we do not pick any particular architecture for the adversarial detector. Instead, we experiment with a variety of models to test their suitability for the task. In the same spirit, we test our setting on different target classifiers, types of attacks, and datasets.

3.3 Experimental Setup

To test our pipeline, four popular classification benchmarks were used: *IMDb* (Maas et al., 2011), *Rotten Tomatoes Movie Reviews* (RTMR) (Pang and Lee, 2005), *Yelp Polarity* (YELP) (Zhang et al., 2015), and *AG News* (Zhang et al., 2015). The first three are binary sentiment analysis tasks in which reviews are classified in either *positive* or *negative* sentiment. The last one, instead, is a classification task where news articles should be identified as one of four possible topics: *World*, *Sports*, *Business*, and *Sci/Tech*.

As main target model for the various tasks we use DistilBERT (Sanh et al., 2020) fine-tuned on IMDb. We choose DistilBert—a transformer language model (Vaswani et al., 2017)—as transformer architectures are widely used in NLP applications, established as state of the art in several tasks, and generally quite resilient to adversarial attacks (Morris et al., 2020). Furthermore, we employ a *Convolutional Neural Network* (CNN) (Zhang et al., 2015), a *Long Short-Term Memory* (LSTM) (Hochreiter and Schmidhuber, 1997), and a full BERT model (Devlin et al., 2019) to test transferability to different target architectures. All models are provided by the TextAttack library (Morris et al., 2020) and are already trained[1] on the datasets used in the experiments.

We generate adversarial text attacks via four well-established word-substitution-based techniques: *Probability Weighted Word Saliency* (PWWS) (Ren et al., 2019), *Improved Genetic Algorithm* (IGA) (Jia et al., 2019), *TextFooler* (Jin et al., 2020), and *BERT-based Adversarial Examples* (BAE) (Garg and Ramakrishnan, 2020). The first is a greedy algorithm that uses word saliency

[1] textattack.readthedocs.io/en/latest/3recipes/models.html, released under MIT License

and prediction probability to determine the word replacement order (Ren et al., 2019). IGA, instead, crafts attacks via mutating sentences and promoting the new ones that are more likely to cause a change in the output. TextFooler ranks words by importance and then replaces the ones with the highest ranks. Finally, BAE, leverages a BERT language model to replace tokens based on their context (Garg and Ramakrishnan, 2020). All attacks are generated using the TextAttack library (Morris et al., 2020).

We investigate several combinations of datasets, target models, and attacks to test our detector in a variety of configurations. Because of its robustness and well-balanced behavior, we pick the average F1-score as our main metric for detection. However, as in adversarial detection false negatives can have major consequences, we also report the recall on adversarial sentences. Later on, in 4.3, we also compare performance with other metrics such as precision and original recall and observe how they are influenced by the chosen decision threshold.

4 Experimental Results

In this section, we report the experimental results of our work. In 4.1, we study various detector architectures to choose the best performing one for the remaining experiments. In 4.2, we measure our pipeline's performance in several configurations (target model, dataset, attack) and we compare it to the current state-of-the-art adversarial detectors. While doing so, we also assess transferability via observing the variation in performance when changing the dataset, the target model, and the attack source without retraining our detector. Finally, in 4.3, we look at how different decision boundaries affect performance metrics.

4.1 Choosing a Detector Model

The proposed method does not impose any constraint on which detector architecture should be used. For this reason, no particular model has been specified in this work so far. We study six different detector architectures in one common setting. We do so in order to pick one to be utilized in the rest of the experiments. Specifically, we compare XGBoost (Chen and Guestrin, 2016), AdaBoost (Schapire, 1999), LightGBM (Ke et al., 2017), SVM (Hearst et al., 1998), Random Forest (Breiman, 2001), and a Perceptron NN (Singh and Banerjee, 2019). All models are compared on adversarial attacks generated with PWWS from IMDb samples and targeting a DistilBERT model fine-tuned on IMDb. A balanced set of $3,000$ instances—$1,500$ normal and $1,500$ adversarial—was used for training the detectors while the test set contains a total of 1360 samples following the same proportions.

Model	F1-Score	Adv. Recall
XGBoost	**92.4**	95.2
AdaBoost	91.8	**96.0**
LightGBM	92.0	93.7
SVM	92.0	94.8
Random Forest	91.5	93.7
Perceptron NN	90.4	88.1

Table 2: Performance comparison of different detector architectures on IMDb adversarial attacks generated with PWWS and targeting a DistilBERT transformer.

As shown in Table 2, all architectures achieve competitive performance and none of them clearly appears superior to the others. We pick XGBoost (Chen and Guestrin, 2016) as it exhibits the best F1-score. The main hyperparameters utilized are 29 gradient boosted trees with a maximum depth of 3 and 0.34 as learning rate. We utilize this detector architecture for all experiments in the following sections.

4.2 Detection Performance

Tables 3a and 3b report the detection performance of our method in a variety of configurations. In each table, the first row represents the setting—i.e. combination of target model, dataset, and attack type—in which the detector was trained. The remaining rows, instead, are w.r.t. settings in which we tested the already trained detector without performing any kind of fine-tuning or retraining.

We utilize a balanced training set of size $3,000$ and $2,400$ samples respectively for the detectors trained on IMDb adversarial attacks (Table 3a) and on AG News attacks (Table 3b). All results are obtained using balanced test sets containing 500 samples. The only exceptions are the configurations (DistilBERT, RTMR, IGA) and (DistilBERT, AG News, IGA) which used test sets of size 480 and 446 respectively due to data availability.

To the best of our knowledge, the FGWS method from Mozes et al. (2021) is the best detector avail-

Configuration			*WDR* (Ours)		*FGWS* (Mozes et al., 2021)	
Model	**Dataset**	**Attack**	**F1-Score**	**Adv. Recall**	**F1-Score**	**Adv. Recall**
DistilBERT	IMDb	PWWS	**92.1** ± **0.5**	94.2 ± 1.1	89.5	82.7
LSTM	IMDb	PWWS	**84.1** ± **3.4**	86.8 ± 8.5	80.0	69.6
CNN	IMDb	PWWS	84.3 ± 3.1	90.0 ± 6.2	**86.3**	79.6
BERT	IMDb	PWWS	**92.4** ± **0.7**	92.5 ± 1.8	89.8	82.7
DistilBERT	AG News	PWWS	**93.1** ± **0.6**	96.1 ± 2.2	89.5	84.6
DistilBERT	RTMR	PWWS	74.1 ± 3.1	85.1 ± 8.6	**78.9**	67.8
DistilBERT	IMDb	TextFooler	**94.2** ± **0.8**	97.3 ± 0.9	86.0	77.6
DistilBERT	IMDb	IGA	**88.5** ± **0.9**	95.5 ± 1.3	83.8	74.8
DistilBERT	IMDb	BAE	**88.0** ± **0.9**	96.3 ± 1.0	65.6	50.2
DistilBERT	RTMR	IGA	**70.4** ± **5.5**	90.2 ± 6.9	68.1	55.2
DistilBERT	RTMR	BAE	**68.5** ± **4.3**	82.2 ± 9.0	29.4	18.5
DistilBERT	AG News	BAE	**81.0** ± **4.3**	95.4 ± 3.8	55.8	44.0
BERT	YELP	PWWS	89.4 ± 0.6	85.3 ± 1.7	**91.2**	85.6
BERT	YELP	TextFooler	**95.9** ± **0.3**	97.5 ± 0.6	90.5	84.2

(a) Performance results for detector trained on (DistilBERT, IMDb, PWWS).

Configuration			*WDR* (Ours)		*FGWS* (Mozes et al., 2021)	
Model	**Dataset**	**Attack**	**F1-Score**	**Adv. Recall**	**F1-Score**	**Adv. Recall**
DistilBERT	AG News	PWWS	**93.6** ± **1.5**	94.8 ± 2.4	89.5	84.6
LSTM	AG News	PWWS	**94.0** ± **1.0**	94.2 ± 2.2	88.9	84.9
CNN	AG News	PWWS	**91.1** ± **1.4**	91.2 ± 2.6	90.6	87.6
BERT	AG News	PWWS	**92.5** ± **0.9**	93.0 ± 1.8	88.7	83.2
DistilBERT	IMDB	PWWS	**91.4** ± **0.6**	93.0 ± 1.9	89.5	82.7
DistilBERT	RTMR	PWWS	75.8 ± 0.9	78.5 ± 4.8	**78.9**	67.8
DistilBERT	AG News	TextFooler	**95.7** ± **0.7**	97.3 ± 1.2	87.0	79.4
DistilBERT	AG News	BAE	**86.4** ± **1.1**	94.5 ± 1.8	55.8	44.0
DistilBERT	AG News	IGA	**86.7** ± **1.5**	93.6 ± 2.1	68.6	58.3
DistilBERT	RTMR	IGA	**73.7** ± **1.5**	85.4 ± 5.2	68.1	55.2
DistilBERT	RTMR	BAE	**71.0** ± **1.1**	75.2 ± 6.0	29.4	18.5
DistilBERT	IMDB	BAE	**88.1** ± **0.9**	97.0 ± 1.0	65.6	55.2
BERT	YELP	PWWS	86.2 ± 1.4	77.2 ± 3.1	**91.2**	85.6
BERT	YELP	TextFooler	**95.4** ± **0.3**	94.7 ± 0.9	90.5	84.2

(b) Performance results for detector trained on (DistilBERT, AG News, PWWS).

Table 3: Adversarial detection performance of our defense against the state of the art *FGWS* under several setups. Results were obtained with a detector trained on two different configurations as indicated in the first row of each table. For all other rows, i.e. test configurations, differences w.r.t the training setup have been highlighted. To increase the results' statistical significance, we average the performance across 30 different data-splits of the training configuration. Additionally, we report the corresponding 95% confidence intervals. Given the deterministic nature of *FGWS*, different data-splits lead to the same performance and hence confidence intervarls are not reported as they are trivial (±0).

able and was already proven to be better than DISP (Zhou et al., 2019) by its authors. Hence, we utlize FGWS as baseline for comparison in all configurations. Analogously to our method, FGWS is trained on the configuration in the first row of each table and then applied to all others. More in detail, we fine-tune its *frequency substitution threshold* parameter δ (Mozes et al., 2021) until achieving a best fit value of $\delta = 0.9$ in both training settings.

From what can be seen in both tables, the proposed method consistently shows very competitive results in terms of F1-score and outperforms the baseline in 22 configurations out of 28 (worse in 5) and is on average better by 8.96 percentage points. At the same time, our methods exhibits a very high adversarial recall, showing a strong capability at identifying attacks and thus producing a small amount of false negatives.

Generalization to different target models: Starting from the training configurations, we vary the *target model* while maintaining the other components fixed (rows 2-4 of each table). Here, the detector achieves state-of-the-art results in all test settings, occasionally dropping below the 90% F1-score on a few simpler models like LSTM and CNN while not exhibiting any decay on more complex models like BERT.

Generalization to different datasets: Analogous to the previous point, we systematically substitute the *dataset* component for evaluation (rows 5-6 of each table). We notice a substantial decay in F1-score when testing with RTMR (74.1 - 75.8%) since samples are short and, therefore, may contain few words which are very relevant for the prediction, just like adversarial replacements. Nevertheless, removing adversarial words still result in a change of prediction to the original class thereby preserving high adversarial recall."

Generalization to different attacks: Results highlight a good reaction to all other text attacks (rows 7-9 of each table) and even experiences a considerable boost in performance against TextFooler. In contrast, the baseline *FGWS* significantly suffers against more complex attacks such as BAE, which generates context-aware perturbation.

Besides testing generalization properties via systematically varying one configuration component at the time, we also test on a few settings presenting changes in multiple ones (rows 10-14 of each table). Also in these settings, the proposed method maintains a very competitive performance, with noticeable drops only on the RTMR dataset.

4.3 Tuning the Decision Boundary

Depending on the application in which the detector is used to monitor the model and detect malicious input manipulations, different performance metrics can be taken into account to determine whether it is safe to deploy the model. For instance, in a very safety-critical application where successful attacks lead to harmful consequences, *adversarial recall* becomes considerably more relevant as a metric than the F1-score.

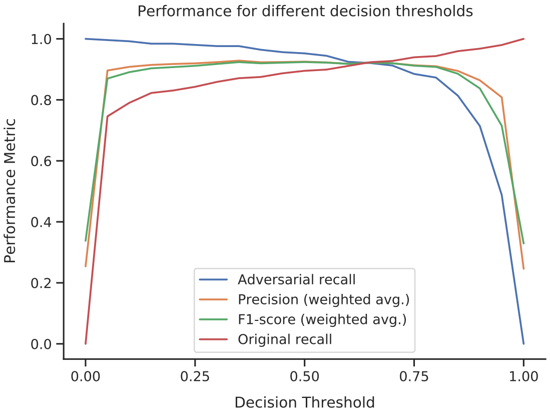


Figure 2: Performance metrics w.r.t. different decision thresholds for our XGBoost classifier on the configuration (IMDb, DistilBERT, PWWS). Input sentences are classified as adversarial when their probability is higher than the decision threshold.

We examine how relevant metrics change in response to different choices for the discrimination threshold. Please note that a lower value corresponds to more caution, i.e. we are more likely to output that a certain input is adversarial.

DT	Precision	F1	Adv. Recall	Orig. Recall
0.5	92.5	92.4	95.2	89.5
0.4	92.3	92.0	96.4	87.5
0.3	92.4	91.8	97.6	85.9
0.15	91.5	90.3	**98.4**	82.3

Table 4: Performance comparison using different *Decision Thresholds* (DT) for our XGBoost classifier on the configuration (IMDb, DistilBERT, PWWS). The used default value is 0.5.

Figure 2 and Table 4 show performance results w.r.t. different threshold choices. We notice that decreasing its value from 0.5 to 0.15 can increase the adversarial recall to over 98% at a small cost in terms of precision and F1-score ($<$ 2 percentage points). Applications where missing attacks—i.e. false negatives—have disastrous consequences could take advantage of this property and consider lowering the decision boundary. This is particularly true if attacks are expected with a low frequency and an increase in false positive incurs only minor costs.

5 Discussion and Qualitative Results

Section 4 discussed quantitative results and emphasized the competitive performance that the proposed approach achieves. Here, instead, we focus on the qualitative aspects of our research findings. For instance, we try to understand *why* our pipeline works while also discussing challenges, limitations, ethical concerns, and future work.

5.1 Understanding the Adversarial Detector

The proposed pipeline consists of a machine learning classifier—e.g. XGBoost—fed with the model's WDR scores. The intuition behind the approach is that words replaced by adversarial attacks play a big role in altering the target model's decision. Despite the competitive detection performance, the detector is itself a learning algorithm and we cannot determine with certainty what patterns it can identify.

To validate our original hypothesis, we apply a popular explainability technique—SHAP (Lundberg and Lee, 2017)—to our detector. This allows us to summarize the effect of each feature at the dataset level. We use the official implementation[2] to estimate the importance of each WDR and use a *beeswarm plot* to visualize the results.

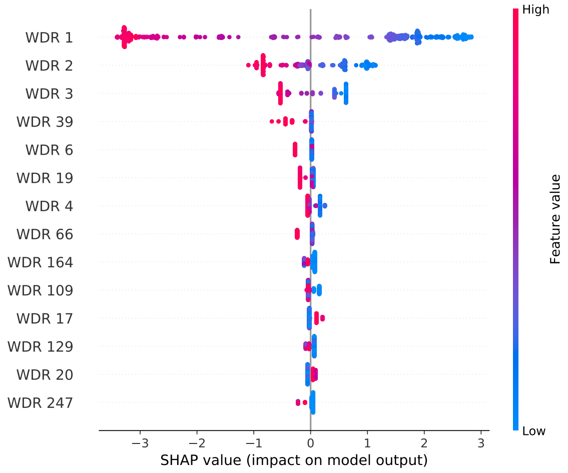


Figure 3: WDR scores with the highest impact (SHAP value) on the detector's prediction. Please recall that the WDR scores are sorted by magnitude. For instance, WDR 1 is the first and largest WDR score.

Figure 3 shows that values in the first positions—i.e. 1, 2, and 3—of the input sequence are those influencing the adversarial detector the most. Since in our pipeline WDR scores are sorted based on their magnitude, this means that the largest WDR of each prediction are the most relevant for the detector. This is consistent with our hypothesis that replaced words substantially change output logits and thus measuring their variation is effective for detecting input manipulations. As expected, negative values for the WDR correspond to a higher likelihood of the input being adversarial.

We also notice that features after the first three do not appear in the naturally expected order. We believe this is the case as for most sentences it is sufficient to replace two-three words to generate an adversarial sample. Hence, in most cases, only a few WDR scores carry important signals for detection.

5.2 Challenges and Limitations

While WDR scores contain rich patterns to identify manipulated samples, they are also relatively expensive to compute. Indeed, we need to run the model once for each feature—i.e. each word—in the input text. While this did not represent a limitation for our use-cases and experiments, we acknowledge that it could result in drawbacks when input texts are particularly long.

Our method is specifically designed against word-level attacks and it does not cover character-level ones. However, the intuition seems to some extent applicable also to sentences with typos and similar artifacts as the words containing them will play a big role for the prediction. This, like in the word-level case, needs to happen in order for the perturbations to result in a successful adversarial text attack and change the target model's prediction

5.3 Ethical Perspective and Future Work

Detecting—or in general defending against—adversarial attacks is a fundamental pillar to deploy machine learning models ethically and safely. However, while defense strategies increase model robustness, they can also inspire and stimulate new and improved attack techniques. An example of this phenomenon is BAE (Garg and Ramakrishnan, 2020), which leverages architectures more resilient to attacks such as BERT to craft highly-effective contextual attacks. Analogously, defense approaches like ours could lead to new attacks that do not rely on a few words to substantially affect output logits.

[2] https://github.com/slundberg/shap, released under MIT License

Based on our current findings, we identify a few profitable directions for future research. **(1)** First of all, the usage of logits-based metrics such as the WDR appears to be very promising for detecting adversarial inputs. We believe that a broader exploration and comparison of other metrics previously used in computer vision could lead to further improvements. **(2)** We encourage future researchers to draw inspiration from this work and also test their defenses in settings that involve mismatched attacks, datasets, and target models. At the same time, we set as a priority for our future work to also evaluate the efficacy of adversarial detection methods on adaptive attacks (Tramer et al., 2020; Athalye et al., 2018). **(3)** This work proves the efficacy of WDR in a variety of settings, which include a few different datasets and tasks. However, it would be beneficial for current research to understand how these techniques would apply to high-stakes NLP applications such as hate speech detection (Mosca et al., 2021; Wich et al., 2021).

6 Conclusion

Adversarial text attacks are a major obstacle to the safe deployment of NLP models in high-stakes applications. However, although manipulated and original samples appear indistinguishable, interpreting the model's reaction can uncover helpful signals for adversarial detection.

Our work utilizes logits of original and adversarial samples to train a simple machine learning detector. WDR scores are an intuitive measure of word relevance and are effective for detecting text components having a suspiciously high impact on the output. The detector does not make any assumption on the classifier targeted by the attacks and can be thus considered model-agnostic.

The proposed approach achieves very promising results, considerably outperforming the previous state-of-the-art in word-level adversarial detection. Experimental results also show the detector to possess remarkable generalization capabilities across different target models, datasets, and text attacks without needing to retrain. These include transformer architectures such as BERT and well-established attacks such as PWWS, genetic algorithms, and context-aware perturbations.

We believe our work sets a strong baseline on which future research can build to develop better defense strategies and thus promoting the safe deployment of NLP models in practice. We release our code to the public to facilitate further research and development [3].

References

Jonathan Aigrain and Marcin Detyniecki. 2019. Detecting adversarial examples and other misclassifications in neural networks by introspection. *arXiv preprint arXiv:1905.09186*.

Basemah Alshemali and Jugal Kalita. 2019. Towards mitigating adversarial texts. *International Journal of Computer Applications*, 178(50):1–7.

Moustafa Alzantot, Yash Sharma, Ahmed Elgohary, Bo-Jhang Ho, Mani Srivastava, and Kai-Wei Chang. 2018. Generating natural language adversarial examples. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2890–2896. Association for Computational Linguistics.

Anish Athalye, Logan Engstrom, Andrew Ilyas, and Kevin Kwok. 2018. Synthesizing robust adversarial examples. In *International conference on machine learning*, pages 284–293. PMLR.

Leo Breiman. 2001. Random forests. *Machine Learning*, 45(1):5–32.

Tianqi Chen and Carlos Guestrin. 2016. Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '16, page 785–794. Association for Computing Machinery.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186. Association for Computational Linguistics.

Xinshuai Dong, Anh Tuan Luu, Rongrong Ji, and Hong Liu. 2021. Towards robustness against natural language word substitutions. In *9th International Conference on Learning Representations (ICLR)*.

Javid Ebrahimi, Anyi Rao, Daniel Lowd, and Dejing Dou. 2018. 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. Association for Computational Linguistics.

Steffen Eger, Gözde Gül Şahin, Andreas Rücklé, Ji-Ung Lee, Claudia Schulz, Mohsen Mesgar, Krishnkant Swarnkar, Edwin Simpson, and Iryna

[3] Public repository: https://github.com/javirandor/wdr

Gurevych. 2019. Text processing like humans do: Visually attacking and shielding NLP systems. 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 1634–1647. Association for Computational Linguistics.

G. Fidel, R. Bitton, and A. Shabtai. 2020. When explainability meets adversarial learning: Detecting adversarial examples using shap signatures. In *2020 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8.

Ji Gao, Jack Lanchantin, Mary Lou Soffa, and Yanjun Qi. 2018. Black-box generation of adversarial text sequences to evade deep learning classifiers. In *2018 IEEE Security and Privacy Workshops (SPW)*, pages 50–56.

Siddhant Garg and Goutham Ramakrishnan. 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.

Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. 2014. Explaining and harnessing adversarial examples. In International Conference on Learning Representations (ICLR).

M.A. Hearst, S.T. Dumais, E. Osuna, J. Platt, and B. Scholkopf. 1998. Support vector machines. *IEEE Intelligent Systems and their Applications*, 13(4):18–28.

Dan Hendrycks and Kevin Gimpel. 2016. Early methods for detecting adversarial images. *arXiv preprint arXiv:1608.00530*.

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

Mohit Iyyer, John Wieting, Kevin Gimpel, and Luke Zettlemoyer. 2018. Adversarial example generation with syntactically controlled paraphrase networks. 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 1875–1885. Association for Computational Linguistics.

Robin Jia, Aditi Raghunathan, Kerem Göksel, and Percy Liang. 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. Association for Computational Linguistics.

Di Jin, Zhijing Jin, Joey Tianyi Zhou, and Peter Szolovits. 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.

Harini Kannan, Alexey Kurakin, and Ian Goodfellow. 2018. Adversarial logit pairing. *arXiv preprint arXiv:1803.06373*.

Guolin Ke, Qi Meng, Thomas Finley, Taifeng Wang, Wei Chen, Weidong Ma, Qiwei Ye, and Tie-Yan Liu. 2017. Lightgbm: A highly efficient gradient boosting decision tree. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

Alexey Kurakin, Ian Goodfellow, and Samy Bengio. 2017. Adversarial examples in the physical world. *ICLR Workshop*.

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

Scott M Lundberg and Su-In Lee. 2017. A unified approach to interpreting model predictions. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 4765–4774. Curran Associates, Inc.

Andrew L. Maas, Raymond E. Daly, Peter T. Pham, Dan Huang, Andrew Y. Ng, and Christopher Potts. 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. Association for Computational Linguistics.

John Morris, Eli Lifland, Jin Yong Yoo, Jake Grigsby, Di Jin, and Yanjun Qi. 2020. 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.

Edoardo Mosca, Maximilian Wich, and Georg Groh. 2021. Understanding and interpreting the impact of user context in hate speech detection. In *Proceedings of the Ninth International Workshop on Natural Language Processing for Social Media*, pages 91–102, Online. Association for Computational Linguistics.

Maximilian Mozes, Pontus Stenetorp, Bennett Kleinberg, and Lewis Griffin. 2021. Frequency-guided word substitutions for detecting textual adversarial examples. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 171–186. Association for Computational Linguistics.

Bo Pang and Lillian Lee. 2005. Seeing stars: Exploiting class relationships for sentiment categorization with respect to rating scales. In *Proceedings of the*

43rd Annual Meeting of the Association for Computational Linguistics (ACL'05), pages 115–124. Association for Computational Linguistics.

Tianyu Pang, Chao Du, Yinpeng Dong, and Jun Zhu. 2018. Towards robust detection of adversarial examples. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, NIPS'18, page 4584–4594, Red Hook, NY, USA. Curran Associates Inc.

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

Shuhuai Ren, Yihe Deng, Kun He, and Wanxiang Che. 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. Association for Computational Linguistics.

Kevin Roth, Yannic Kilcher, and Thomas Hofmann. 2019. The odds are odd: A statistical test for detecting adversarial examples. In *International Conference on Machine Learning*, pages 5498–5507. PMLR.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2020. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. *arXiv preprint arXiv:1910.01108*.

Robert E. Schapire. 1999. A brief introduction to boosting. In *Proceedings of the 16th International Joint Conference on Artificial Intelligence - Volume 2*, IJCAI'99, page 1401–1406. Morgan Kaufmann Publishers Inc.

Jaswinder Singh and Rajdeep Banerjee. 2019. A study on single and multi-layer perceptron neural network. In *2019 3rd International Conference on Computing Methodologies and Communication (ICCMC)*, pages 35–40.

Guanhong Tao, Shiqing Ma, Yingqi Liu, and Xiangyu Zhang. 2018. Attacks meet interpretability: Attribute-steered detection of adversarial samples. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, NIPS'18, page 7728–7739. Curran Associates Inc.

Florian Tramer, Nicholas Carlini, Wieland Brendel, and Aleksander Madry. 2020. On adaptive attacks to adversarial example defenses. *Advances in Neural Information Processing Systems*, 33:1633–1645.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

Xiaosen Wang, Hao Jin, and Kun He. 2019. Natural language adversarial attacks and defenses in word level. *arXiv preprint arXiv:1909.06723*.

Yaopeng Wang, Lehui Xie, Ximeng Liu, Jia-Li Yin, and Tingjie Zheng. 2021. Model-agnostic adversarial example detection through logit distribution learning. In *2021 IEEE International Conference on Image Processing (ICIP)*, pages 3617–3621.

Maximilian Wich, Edoardo Mosca, Adrian Gorniak, Johannes Hingerl, and Georg Groh. 2021. Explainable abusive language classification leveraging user and network data. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 481–496. Springer.

Dengpan Ye, Chuanxi Chen, Changrui Liu, Hao Wang, and Shunzhi Jiang. 2020. Detection defense against adversarial attacks with saliency map. *arXiv preprint arXiv:2009.02738*.

Wei Emma Zhang, Quan Z. Sheng, Ahoud Alhazmi, and Chenliang Li. 2020. Adversarial attacks on deep-learning models in natural language processing: A survey. *ACM Trans. Intell. Syst. Technol.*, 11(3).

Xiang Zhang, Junbo Zhao, and Yann LeCun. 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. MIT Press.

Yi Zhou, Xiaoqing Zheng, Cho-Jui Hsieh, Kai-wei Chang, and Xuanjing Huang. 2020. Defense against adversarial attacks in nlp via dirichlet neighborhood ensemble. *arXiv preprint arXiv:2006.11627*.

Yichao Zhou, Jyun-Yu Jiang, Kai-Wei Chang, and Wei Wang. 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. Association for Computational Linguistics.

B

PUBLICATIONS [†]

Publications in Appendix B are not formally relevant for examination in accordance with Exhibit 6 of the regulations for the award of the doctoral degree.

B.1 STUDY II

Edoardo Mosca, Defne Demirtürk, Luca Mülln, Fabio Raffagnato, and Georg Groh (May 2022). "GrammarSHAP: An Efficient Model-Agnostic and Structure-Aware NLP Explainer." In: *Proceedings of the First Workshop on Learning with Natural Language Supervision*. Dublin, Ireland: Association for Computational Linguistics, pp. 10–16. DOI: 10.18653/v1/2022.lnls-1.2. URL: https://aclanthology.org/2022.lnls-1.2

[1] https://creativecommons.org/licenses/by/4.0/

Publication Summary

"Interpreting NLP models is fundamental for their development as it can shed light on hidden properties and unexpected behaviors. However, while transformer architectures exploit contextual information to enhance their predictive capabilities, most of the available methods to explain such predictions only provide importance scores at the word level. This work addresses the lack of feature attribution approaches that also take into account the sentence structure. We extend the SHAP framework by proposing GrammarSHAP—a model-agnostic explainer leveraging the sentence's constituency parsing to generate hierarchical importance scores." (Mosca, Demirtürk, et al., 2022, p. 1)

Author Contributions

Edoardo Mosca contributed to the study as follows:

- Conception, development, and lead of the research project **100%**
- Literature review and feasibility study **80%**
- Setup and implementation of experiments **20%**
- Analysis and interpretation of results. **50%**
- Drafting of the manuscript **90%**
- Submission, peer review, and publication process **100%**

GrammarSHAP: An Efficient Model-Agnostic and Structure-Aware NLP Explainer

Edoardo Mosca
TU Munich,
Department of Informatics,
Germany
edoardo.mosca@tum.de

Defne Demitürk
TU Munich,
Department of Informatics,
Germany
ge75yod@mytum.de

Luca Mülln
TU Munich,
Department of Informatics,
Germany
luca.muelln@tum.de

Fabio Raffagnato
TU Munich,
Department of Informatics,
Germany
ga24giv@mytum.de

Georg Groh
TU Munich,
Department of Informatics,
Germany
grohg@in.tum.de

Abstract

Interpreting NLP models is fundamental for their development as it can shed light on hidden properties and unexpected behaviors. However, while transformer architectures exploit contextual information to enhance their predictive capabilities, most of the available methods to explain such predictions only provide importance scores at the word level. This work addresses the lack of feature attribution approaches that also take into account the sentence structure. We extend the SHAP framework by proposing GrammarSHAP—a model-agnostic explainer leveraging the sentence's constituency parsing to generate hierarchical importance scores.

1 Introduction

Deep learning models have raised the bar in terms of performance in a variety of *Natural Language Processing* (NLP) tasks (Vaswani et al., 2017; Devlin et al., 2019). However, also model complexity has been steadily increasing, which in turn hinders the interpretability of their predictions. This is particularly true for transformer architectures, currently established as the state of the art in various applications but at the same time containing billions of parameters (Brown et al., 2020).

Local explanations have become a popular tool to understand and interpret models' decisions (Madsen et al., 2021; Arrieta et al., 2020). These—besides increasing the public's trust in machine learning systems—can uncover unwanted behaviors such as unintended bias (Madsen et al., 2021; Dixon et al., 2018).

Feature attribution explanations are the most commonly used and can highlight parts of the input text that are relevant for the obtained outcome (Lundberg and Lee, 2017; Ribeiro et al., 2016). Almost all available methods, however, can only attribute a relevance score to single words. This is highly unintuitive as natural language in human communication can be very articulated and context-dependent. Indeed, a word's neighborhood can drastically alter its intended message and sentiment.

Our work focuses on generating explanations that account for the language structure. More specifically, we build hierarchical explanations that attribute relevance scores to sentence constituents at multiple levels. In contrast to previous work addressing the same issue (Chen et al., 2020; Chen and Jordan, 2020), we build our approach as an extension of SHAP (Lundberg and Lee, 2017)—a local explainability framework renowned for its solid theoretical background. Our contribution can be summarized as follows:

(1) We design GrammarSHAP, a model-agnostic approach for generating multi-level explanations that consider the text's structure and its constituents. More specifically, a constituency parsing layer for multi-word tokens selection is added before an adapted KernelSHAP explainer.

(2) We propose to drop the SHAP standard background dataset and use masking tokens instead. This reduces unwanted artifacts in the generated explanations and speeds up the approach's run time.

Proceedings of the First Workshop on Learning with Natural Language Supervision, pages 10 - 16
May 26, 2022 ©2022 Association for Computational Linguistics

(3) We qualitatively compare our method to existing ones in terms of explanation quality and necessary computational effort.

2 Related Work

Several local explainability techniques exist to interpret predictions produced by NLP models (Arrieta et al., 2020). Among them, *features attribution* (or *feature relevance*) approaches quantify each input component's contribution to the model's output, i.e. how each feature affects the observed prediction. Methods in this category are available in a large variety: gradient-based (Simonyan et al., 2014; Sundararajan et al., 2017), neural-network specific e.g. LRP (Bach et al., 2015) and DeepLIFT (Shrikumar et al., 2017), and model-agnostic e.g. LIME (Ribeiro et al., 2016). SHAP (Lundberg and Lee, 2017)—particularly relevant for our methodology—is by many considered to be a gold standard thanks to its solid theoretical background and broad applicability. This framework builds a unified view of methods like LIME, LRP, and DeepLIFT and the game-theoretic concept of Shapley values (Shapley, 1953).

More recent works address the limitations of word-level relevance scores by focusing on phrase-level and hierarchical explanations. The proposed approaches analyze and quantify words' interactions through exhaustive search (Tsang et al., 2018), combining their contextual decomposition scores (Singh et al., 2018), or via measuring SHAP interaction values along a predefined tree structure (Lundberg et al., 2018). Chen and Jordan (2020) combines a linguistic parse tree with Banzhaf values (Banzhaf III, 1964) to capture meaningful interactions in text inputs. (Chen et al., 2020), instead, propose to detect directly feature interaction without resorting to external structures. They propose a hierarchical explainability method that, in a top-down fashion, breaks down text components in shorter phrases and words based on the weakest detected interactions.

3 Methodology

We extend the SHAP framework (Lundberg and Lee, 2017) by proposing a model-agnostic explainer that considers the text's structural dependencies to generate importance scores at multiple levels. In particular, we couple a constituency parsing layer to hierarchically select multi-word tokens with a custom version of KernelSHAP adapted for improved efficiency and run-time. Figure 1 presents an overview of the methodological pipeline proposed in this work.

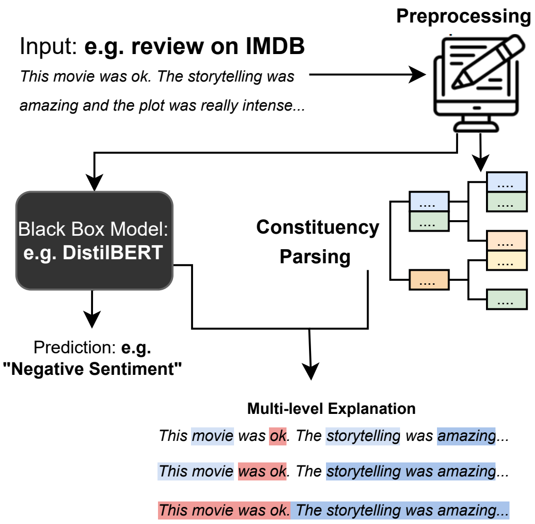

Preprocessing
Input: e.g. review on IMDB
This movie was ok. The storytelling was amazing and the plot was really intense...
Black Box Model: e.g. DistilBERT
Constituency Parsing
Prediction: e.g. "Negative Sentiment"
Multi-level Explanation
This movie was ok. The storytelling was amazing...
This movie was ok. The storytelling was amazing...
This movie was ok. The storytelling was amazing...

Figure 1: Overview of the proposed methodology.

3.1 Token Selection via Constituency Parsing

To hierarchically construct multi-word tokens in a way that reflects the sentence structure, we leverage constituency parsing to group together tokens based on their grammatical interactions. To this end, we choose a state-of-the-art constituency parser: the Berkeley Neural Parser (Kitaev and Klein, 2018).

We iterate over parsed sentences from the single-word level ($depth = 0$) until the complete sentences are grouped up as a single token ($depth = N$). Additionally, we provide a library to retrieve groups of words at any depth, constituents, and combinations thereof. Our implementation also handles inconsistencies between the word-tokenization of the constituency parser and BERT. This is necessary as BERT's tokenizer uses sub-word tokens to represent OOV words and the Berkley Neural Parser[1] only allows full words as input.

3.2 Efficient Multi-Token Explainer

Our GrammarSHAP explainer directly extends the KernelSHAP method from Lundberg and Lee (2017). As parsed sentences already provide a hierarchical structure of grammatically coherent tokens, our extension is not required to compute tokens interaction to construct importance scores for multi-word tokens.

[1] spacy.io/universe/project/self-attentive-parser

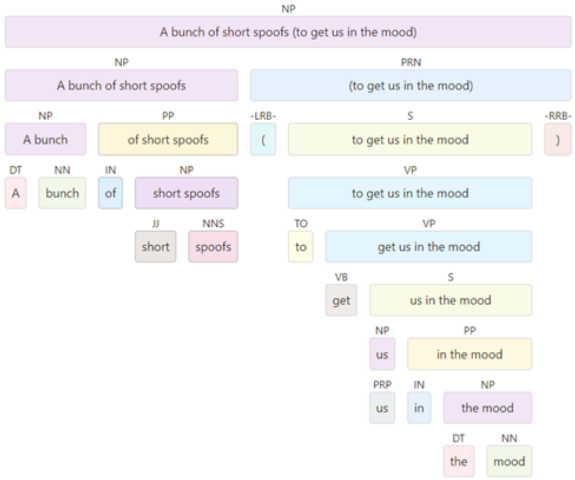


Figure 2: Example of sentence parsed with the Berkeley Neural Parser (Kitaev and Klein, 2018). Tokens are hierarchically grouped from single words (bottom level) to the whole sentence (top level)

KernelSHAP takes an input sample x, a predicting model f, and a background set of samples to be used when replacing tokens to compute feature importance. Tokens belonging to the background dataset are fed to the explainer during initialization. At explanation time, a linear system of all perturbed sentences and their corresponding model predictions is solved to determine the effect of each single feature.

The extension to multi-word tokens consists in feeding the explainer—i.e. KernelSHAP—with the indices corresponding to the features to be grouped. In the case of constituency parsed sentences, indices representing multi-token groups are always adjacent in the input sentence. However, this is not a strict requirement for the following steps of our extension. To obtain group-level feature importance, we constrain the extended explainer to always replace a complete group of words with elements of the background dataset. Analogous to KernelSHAP, the expected effect of each feature group—i.e. its (multi-token) SHAP value—is calculated by solving the linear system of all perturbed sentences with their corresponding outcomes. In summary, our extension behaves like KernelSHAP but treats groups of tokens as single features.

While the calculation of SHAP values on multi-words tokens is a straightforward extension, it leads to several issues:

- **Computationally Expensive:** Computing importance scores for multiple levels further slows down the already inefficient KernelSHAP.
- **Unidirectional:** The explainer only highlights groups with the same sentiment as the overall sentence.
- **High Attribution for [SEP]:** The separation token changes the sentence length when used as replacement from the background data. This causes it to have high relevance for the classifier.

We address these limitation by replacing the background data with **[MASK]** tokens. This leads to a 60-folds speed up of the explainer that is not required to iterate over the background data. Moreover, **[SEP]** does causes explanation artifacts as it is excluded from the background data.

4 Empirical Findings

4.1 Data and Model to be Explained

To test and compare our method in practice, we pick a DistilBERT model (Sanh et al., 2019). Our choice is motivated by transformer architectures being established as the current state of the art in a variety of NLP applications.

Concerning the data, we pick the IMDb movie reviews (Maas et al., 2011) and the SST-2 datasets (Socher et al., 2013). For both, the *Hugging Face*[2] library provides a version of DistilBERT pre-trained on the task of binary sentiment analysis. The accuracy achieved is 93.7% and 91.3% respectively.

4.2 Existing SHAP Baselines

We compare explanations generated with GrammarSHAP with two existing baselines from the SHAP framework (Lundberg and Lee, 2017):

(1) PartitionSHAP, i.e. the library's current recommended method for sentiment analisys on text data. Similarly to our method, it also utilizes **[MASK]** tokens for efficient word removal. However, features are only grouped via a binary tree and thus only token pairs are considered at a given hierarchical level.

(2) KernelSHAP, i.e. the library's standard for model-agnostic explanations. KernelSHAP only produces word-level explanations by default. But thanks to the additive nature of Shapley values,

[2] https://huggingface.co/textattack/distilbert-base-uncased-imdb

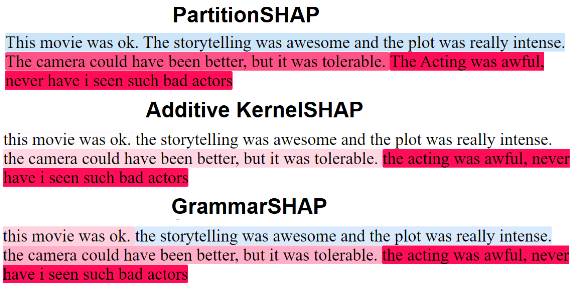
these can be added together according to the constituency parsing tree. We will refer to this custom hierarchical version of KernelSHAP as *Additive KernelSHAP*.

4.3 Comparison

The three methods substantially differ both in terms of generation times and explanation quality. Table 1 reports the average running time to produce an explanation. Figures 3 and 4 show—starting from the same input text—the explanations generated with each method. The text sample is particularly instructive as it contains both positive- and negative-sentiment sentences.

Method	Running Time
PartitionSHAP	2
Add. KernelSHAP	3554 (∼1h)
GrammarSHAP	183 (∼3min)

Table 1: Average running time (in seconds) for GrammarSHAP compared to the existing SHAP baselines. The running time has been measured on 20 randomly selected samples (10 from IMDb and 10 from SST-2). Results were measured on a laptop machine: AMD Ryzen 5 CPU, Nvidia GPU GeForce GTX 1650, 16 GB DDR4 RAM.


PartitionSHAP

This movie was ok. The storytelling was awesome and the plot was really intense. The camera could have been better, but it was tolerable. The Acting was awful, never have i seen such bad actors

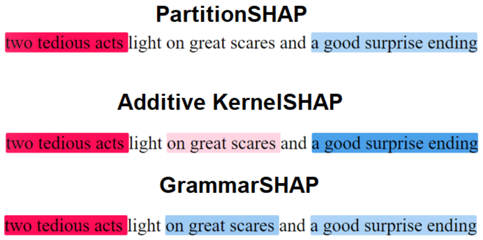
Additive KernelSHAP

this movie was ok. the storytelling was awesome and the plot was really intense. the camera could have been better, but it was tolerable. the acting was awful, never have i seen such bad actors

GrammarSHAP

this movie was ok. the storytelling was awesome and the plot was really intense. the camera could have been better, but it was tolerable. the acting was awful, never have i seen such bad actors

Figure 3: Comparison of three explanation methods for grouped features relevance (5th level). DistilBERT predicted the sample's sentiment as negative with a 79.5% confidence.


PartitionSHAP

two tedious acts light on great scares and a good surprise ending

Additive KernelSHAP

two tedious acts light on great scares and a good surprise ending

GrammarSHAP

two tedious acts light on great scares and a good surprise ending

Figure 4: Comparison of three explanation methods for grouped features relevance (5th level). DistilBERT predicted the sample's sentiment as negative with a 81.8% confidence.

PartitionSHAP is very efficient and the fastest method among the compared ones. However, it is quite coarse in grouping together tokens and fails to identify fine-grained contributions at the sub-sentence level. Additive KernelSHAP has an extremely long execution time and is the slowest of the three approaches. Moreover, it does not identify contributions opposite to the sample's overall sentiment. In contrast, GrammarSHAP is able to identify both negative and positive sentiments at different (hierarchical) levels of granularity. In terms of efficiency, GrammarSHAP does not match the performance of PartitionSHAP. However, its running time is still reasonable and does not raise issues for most applications.

More examples of hierarchical GrammarSHAP explanation on (long) texts are provided in the appendix (see A). There, we also focus on presenting the explanations at different levels of granularity.

5 Limitations and Future Work

GrammarSHAP meaningfully extends the SHAP framework by providing efficient hierarchical explanations that reflect the sentence structure. However, limitations of our methodology and experimentation need to be acknowledged and motivate our future work.

Regarding the explanation quality, our evaluation process is based on the introduced methodological improvements and on a qualitative analysis of the produced explanations. Although evaluation metrics for explanations are complex to define and have not been standardized yet, our comparison would considerably benefit from the usage of quantitative diagnostic properties (Atanasova et al., 2020) and word-level level metrics (Nguyen, 2018; Samek et al., 2016).

In terms of execution time, our method is still reasonable considering the granularity of contributions that it can detect. However, the necessity for further improvements in terms of efficiency becomes apparent when producing real-time explanations on the large scale.

6 Conclusion

In this work we proposed GrammarSHAP: a model-agnostic explainer for text data that accounts for the sentence structure and the existing grammatical relationships between the text tokens. Our approach leverages constituency parsing to extend the SHAP framework by providing hierarchical explanations that go beyond word-level attribution scores.

Our qualitative analysis of the produced explanation yields promising results as GrammarSHAP appears to identify more fine-grained contribution in structured text than its existing SHAP counterparts. At the same time, the usage of masking tokens instead of a background dataset considerably speeds up its execution in comparison with KernalSHAP. These properties make GrammarSHAP also suitable for long texts, especially if they contain sentences carrying different types of sentiment. As a first priority for our future work, we will focus on the quantitative evaluation the produced explanation.

References

Alejandro Barredo Arrieta, Natalia Díaz-Rodríguez, Javier Del Ser, Adrien Bennetot, Siham Tabik, Alberto Barbado, Salvador García, Sergio Gil-López, Daniel Molina, Richard Benjamins, et al. 2020. Explainable artificial intelligence (xai): Concepts, taxonomies, opportunities and challenges toward responsible ai. *Information Fusion*, 58:82–115.

Pepa Atanasova, Jakob Grue Simonsen, Christina Lioma, and Isabelle Augenstein. 2020. A diagnostic study of explainability techniques for text classification. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 3256–3274.

Sebastian Bach, Alexander Binder, Grégoire Montavon, Frederick Klauschen, Klaus-Robert Müller, and Wojciech Samek. 2015. On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation. *PloS one*, 10(7):130–140.

John F Banzhaf III. 1964. Weighted voting doesn't work: A mathematical analysis. *Rutgers L. Rev.*, 19:317.

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

Hanjie Chen, Guangtao Zheng, and Yangfeng Ji. 2020. Generating hierarchical explanations on text classification via feature interaction detection. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5578–5593.

Jianbo Chen and Michael Jordan. 2020. Ls-tree: Model interpretation when the data are linguistic. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 3454–3461.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *NAACL-HLT (1)*.

Lucas Dixon, John Li, Jeffrey Sorensen, Nithum Thain, and Lucy Vasserman. 2018. Measuring and mitigating unintended bias in text classification. In *Proceedings of the 2018 AAAI/ACM Conference on AI, Ethics, and Society*, pages 67–73.

Nikita Kitaev and Dan Klein. 2018. Constituency parsing with a self-attentive encoder. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2676–2686, Melbourne, Australia. Association for Computational Linguistics.

Scott M Lundberg, Gabriel G Erion, and Su-In Lee. 2018. Consistent individualized feature attribution for tree ensembles. *arXiv preprint arXiv:1802.03888*.

Scott M. Lundberg and Su-In Lee. 2017. A unified approach to interpreting model predictions. *NeurIPS 2017*.

Andrew L. Maas, Raymond E. Daly, Peter T. Pham, Dan Huang, Andrew Y. Ng, and Christopher Potts. 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.

Andreas Madsen, Siva Reddy, and Sarath Chandar. 2021. Post-hoc interpretability for neural nlp: A survey. *arXiv preprint arXiv:2108.04840*.

Dong Nguyen. 2018. Comparing automatic and human evaluation of local explanations for 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 1069–1078.

Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. 2016. " why should i trust you?" explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, pages 1135–1144.

Wojciech Samek, Alexander Binder, Grégoire Montavon, Sebastian Lapuschkin, and Klaus-Robert Müller. 2016. Evaluating the visualization of what a deep neural network has learned. *IEEE transactions on neural networks and learning systems*, 28(11):2660–2673.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. *NeurIPS 2017*.

Lloyd S Shapley. 1953. A value for n-person games. *Contributions to the Theory of Games 2.28*, page 307–317.

Avanti Shrikumar, Peyton Greenside, and Anshul Kundaje. 2017. Learning important features through propagating activation differences. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 3145–3153.

Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. 2014. Deep inside convolutional networks: Visualising image classification models and saliency maps. In *2nd International Conference on Learning Representations, ICLR 2014*.

Chandan Singh, W James Murdoch, and Bin Yu. 2018. Hierarchical interpretations for neural network predictions. In *International Conference on Learning Representations*.

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

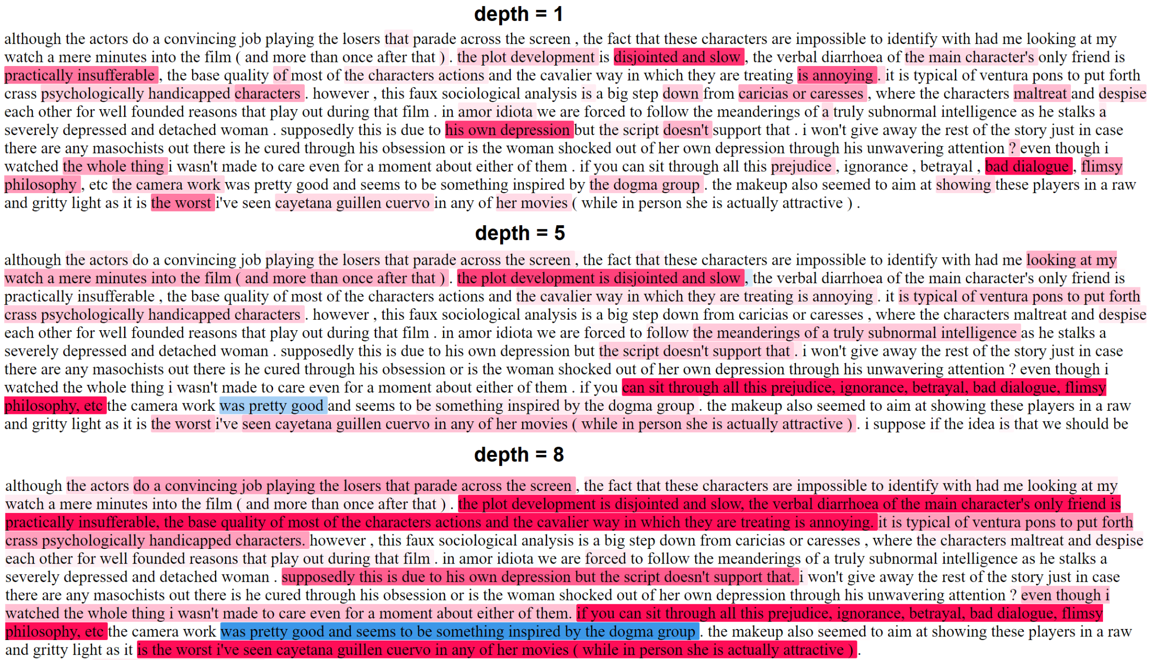
Mukund Sundararajan, Ankur Taly, and Qiqi Yan. 2017. Axiomatic attribution for deep networks. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 3319–3328. JMLR. org.

Michael Tsang, Youbang Sun, Dongxu Ren, and Yan Liu. 2018. Can i trust you more? model-agnostic hierarchical explanations. *arXiv preprint arXiv:1812.04801*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *NIPS 2017*.

A Explanations Examples

Figure 5 shows an example of hierarchical GrammarSHAP explanation on a long text while 6 rather focuses on a shorter text. More examples can be found in the code repository attached to our submission. These are in the *Graphics Interchange Format* (GIF) format to visualize the transformation of the relevance scores through the various hierarchical levels.

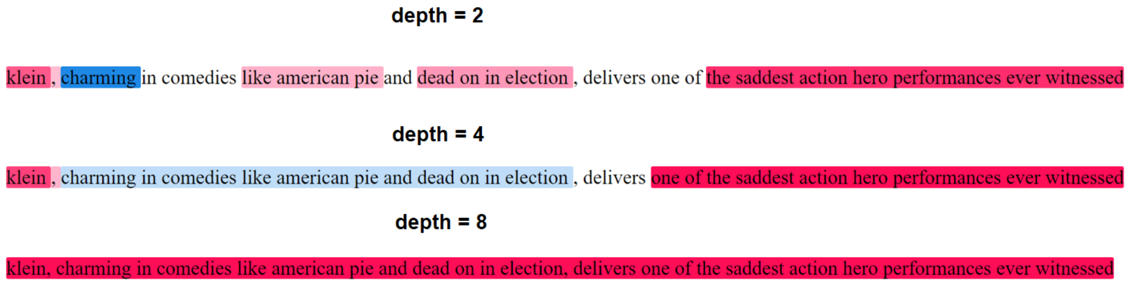


although the actors do a convincing job playing the losers that parade across the screen , the fact that these characters are impossible to identify with had me looking at my watch a mere minutes into the film (and more than once after that) . the plot development is disjointed and slow , the verbal diarrhoea of the main character's only friend is practically insufferable , the base quality of most of the characters actions and the cavalier way in which they are treating is annoying . it is typical of ventura pons to put forth crass psychologically handicapped characters . however , this faux sociological analysis is a big step down from caricias or caresses , where the characters maltreat and despise each other for well founded reasons that play out during that film . in amor idiota we are forced to follow the meanderings of a truly subnormal intelligence as he stalks a severely depressed and detached woman . supposedly this is due to his own depression but the script doesn't support that . i won't give away the rest of the story just in case there are any masochists out there is he cured through his obsession or is the woman shocked out of her own depression through his unwavering attention ? even though i watched the whole thing i wasn't made to care even for a moment about either of them . if you can sit through all this prejudice , ignorance , betrayal , bad dialogue , flimsy philosophy , etc the camera work was pretty good and seems to be something inspired by the dogma group . the makeup also seemed to aim at showing these players in a raw and gritty light as it is the worst i've seen cayetana guillen cuervo in any of her movies (while in person she is actually attractive) .

although the actors do a convincing job playing the losers that parade across the screen , the fact that these characters are impossible to identify with had me looking at my watch a mere minutes into the film (and more than once after that) . the plot development is disjointed and slow, the verbal diarrhoea of the main character's only friend is practically insufferable , the base quality of most of the characters actions and the cavalier way in which they are treating is annoying . it is typical of ventura pons to put forth crass psychologically handicapped characters . however , this faux sociological analysis is a big step down from caricias or caresses , where the characters maltreat and despise each other for well founded reasons that play out during that film . in amor idiota we are forced to follow the meanderings of a truly subnormal intelligence as he stalks a severely depressed and detached woman . supposedly this is due to his own depression but the script doesn't support that . i won't give away the rest of the story just in case there are any masochists out there is he cured through his obsession or is the woman shocked out of her own depression through his unwavering attention ? even though i watched the whole thing i wasn't made to care even for a moment about either of them . if you can sit through all this prejudice, ignorance, betrayal, bad dialogue, flimsy philosophy, etc the camera work was pretty good and seems to be something inspired by the dogma group . the makeup also seemed to aim at showing these players in a raw and gritty light as it is the worst i've seen cayetana guillen cuervo in any of her movies (while in person she is actually attractive) . i suppose if the idea is that we should be

although the actors do a convincing job playing the losers that parade across the screen , the fact that these characters are impossible to identify with had me looking at my watch a mere minutes into the film (and more than once after that) . the plot development is disjointed and slow, the verbal diarrhoea of the main character's only friend is practically insufferable, the base quality of most of the characters actions and the cavalier way in which they are treating is annoying. it is typical of ventura pons to put forth crass psychologically handicapped characters. however , this faux sociological analysis is a big step down from caricias or caresses , where the characters maltreat and despise each other for well founded reasons that play out during that film . in amor idiota we are forced to follow the meanderings of a truly subnormal intelligence as he stalks a severely depressed and detached woman . supposedly this is due to his own depression but the script doesn't support that. i won't give away the rest of the story just in case there are any masochists out there is he cured through his obsession or is the woman shocked out of her own depression through his unwavering attention ? even though i watched the whole thing i wasn't made to care even for a moment about either of them. if you can sit through all this prejudice, ignorance, betrayal, bad dialogue, flimsy philosophy, etc the camera work was pretty good and seems to be something inspired by the dogma group . the makeup also seemed to aim at showing these players in a raw and gritty light as it is the worst i've seen cayetana guillen cuervo in any of her movies (while in person she is actually attractive).

Figure 5: Explanation generated with GrammarSHAP on a long IMDB review with negative-sentiment prediction of 91.7%. From top to bottom, relevance scores at the 1st, 5th and 8th hierarchical level.



depth = 2

klein , charming in comedies like american pie and dead on in election , delivers one of the saddest action hero performances ever witnessed

depth = 4

klein , charming in comedies like american pie and dead on in election , delivers one of the saddest action hero performances ever witnessed

depth = 8

klein, charming in comedies like american pie and dead on in election, delivers one of the saddest action hero performances ever witnessed

Figure 6: Explanation generated with GrammarSHAP on a short SST-2 review with negative-sentiment prediction of 91.6%. From top to bottom, relevance scores at the 2nd, 4th and 8th hierarchical level.

B.2 STUDY IV

Reprinted with permission from:

Maximilian Wich, Edoardo Mosca, Adrian Gorniak, Johannes Hingerl, and Georg Groh (2021). “Explainable abusive language classification leveraging user and network data.” In: *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*. Springer, pp. 481–496. URL: https://2021.ecmlpkdd.org/wp-content/uploads/2021/07/sub_663.pdf

Publication Summary

"Online hate speech is a phenomenon with considerable consequences for our society. Its automatic detection using machine learning is a promising approach to contain its spread. However, classifying abusive language with a model that purely relies on text data is limited in performance due to the complexity and diversity of speech (e.g., irony, sarcasm). Moreover, studies have shown that a significant amount of hate on social media platforms stems from online hate communities. Therefore, we develop an abusive language detection model leveraging user and network data to improve the classification performance. We integrate the explainable AI framework SHAP (SHapley Additive exPlanations) to alleviate the general issue of missing transparency associated with deep learning models, allowing us to assess the model's vulnerability toward bias and systematic discrimination reliably. Furthermore, we evaluate our multimodel architecture on three datasets in two languages (i.e., English and German). Our results show that user-specific timeline and network data can improve the classification, while the additional explanations resulting from SHAP make the predictions of the model interpretable to humans." (Wich, Mosca, et al., 2021, p. 1)

Author Contributions

Edoardo Mosca contributed to the study as follows:

- Conception, development, and lead of the research project **30%**
- Literature review and feasibility study **30%**
- Methodology and experimental design **20%**
- Implementation and interpretation of results. **20%**
- Drafting of the manuscript **20%**
- Submission, peer review, and publication process **0%**

Explainable Abusive Language Classification Leveraging User and Network Data

Maximilian Wich[0000−0002−9149−9454] ✉, Edoardo Mosca[0000−0003−4045−5328], Adrian Gorniak[0000−0002−6165−5807], Johannes Hingerl[0000−0002−8260−032X], and Georg Groh[0000−0002−5942−2297]

Technical University of Munich, Germany
{maximilian.wich,edoardo.mosca,adrian.gorniak,johannes.hingerl}@tum.de
grohg@in.tum.de

Abstract. Online hate speech is a phenomenon with considerable consequences for our society. Its automatic detection using machine learning is a promising approach to contain its spread. However, classifying abusive language with a model that purely relies on text data is limited in performance due to the complexity and diversity of speech (e.g., irony, sarcasm). Moreover, studies have shown that a significant amount of hate on social media platforms stems from online hate communities. Therefore, we develop an abusive language detection model leveraging user and network data to improve the classification performance. We integrate the explainable AI framework SHAP (SHapley Additive exPlanations) to alleviate the general issue of missing transparency associated with deep learning models, allowing us to assess the model's vulnerability toward bias and systematic discrimination reliably. Furthermore, we evaluate our multimodel architecture on three datasets in two languages (i.e., English and German). Our results show that user-specific timeline and network data can improve the classification, while the additional explanations resulting from SHAP make the predictions of the model interpretable to humans.

Keywords: Hate speech · Abusive language · Classification model · Social network · Deep learning · Explainable AI

Warning: This paper contains content that may be abusive or offensive.

1 Introduction

Hate speech is a severe challenge that social media platforms such as Twitter and Facebook face nowadays. However, it is not purely an online phenomenon and can spill over to the offline world resulting in physical violence [36]. The Capitol riots in the US at the beginning of the year are a tragic yet prime example. Therefore, the fight against hate speech is a crucial societal challenge.

The enormous amount of user-generated content excludes manual monitoring as a viable solution. Hence, automatic detection of hate speech becomes the key component of this challenge. A technology to facilitate the identification is *Machine Learning*. Especially in recent years, *Natural Language Processing* (NLP) has made significant progress. Even if these advances also enhanced hate speech classification models, there is room for improvement [29].

However, gaining the last points of the F1 score is a massive challenge in the context of hate speech. Firstly, abusive language has various forms, types, and targets [32]. Secondly, language itself is a complex and evolving construct; e.g., a word can have multiple meanings, people create new words or use them differently [29]. This complexity exacerbates classifying abusive language purely based on textual data. Therefore, researchers have started to look beyond pure text-driven classification and discovered the relevance of social network data [10]. Kreißel et al. [11], for example, showed that small subnetworks cause a significant portion of offensive and hateful content on social media platforms. Thus, it is beneficial to integrate network data into the model [3, 22, 15, 5, 6]. However, to the best of our knowledge, no one has investigated the impact of combining the text data of the post that is meant to be classified, the user's previous posts, and their social network data.

An issue with such an approach is its vulnerability to bias, meaning that a system "systematically and unfairly discriminate[s] against certain individuals or groups of individuals in favor of others" [7, p. 332]. *Deep Learning* (DL) models often used in NLP are particularly prone to this issue because of their black-box nature [17]. Conversely, a system combining various data sources and leveraging user-related data has a more considerable potential of discriminating individuals or groups. Consequently, such systems should integrate *eXplainable AI* (XAI) techniques to address this issue and increase trustworthiness.

We address the following two research questions in our paper concerning the two discussed aspects:

RQ1 Can abusive language classification be improved by leveraging users' previous posts and their social network data?

RQ2 Can explainable AI be used to make predictions of a multimodal hate speech classification model more understandable?

To answer the research questions, we develop an explainable multimodal classification model for abusive language using the mentioned data sources[1]. We evaluate our model on three different datasets—WASEEM [33], DAVIDSON [4], and WICH [35]. Furthermore, we report findings of integrating user and social network data that are relevant for future work.

2 Related Work

Most work in the abusive language detection domain has focused on developing models that only use the text data of the document to be classified [29, 16, 24].Other works, however, have started to integrate context-related data into abusive language detection [29, 24, 18]. One promising data source is the users'

[1] Code available on https://github.com/mawic/multimodal-abusive-language-detection

social network because it has been shown that hater networks on social media platforms cause a considerable amount of online hate [11, 8]. Combining network and text data from Twitter was already successfully applied to predict whether an account is verified [2] or to identify extremist accounts [38]. In the case of abusive language, Papegnies et al. [19] built a classification model using local and global topological measures from graphs as features for cyberbullying detection (e.g., average distance, betweenness centrality). A similar approach has been applied by Chatzakou et al. [3], but they also integrated user-related data (e.g., number of posts, account age) and textual data (e.g., number of hashtags). This approach was picked up and extended by other researchers [6, 5] (e.g., integrating users' gender, geolocation) who confirmed the usefulness of additional context-related data sources. They all have in common that the network features are only topological measures and do not contain any information about the relations. Mishra et al. [15] addressed this downside and modeled the users' follower network with a node2vec embedding that serves as an additional input for the classification model. Ribeiro et al. [22] developed a similar model; they, however, used the graph embedding GraphSAGE to model the retweet network and combined it with a document embedding for the text data [9]. For this purpose, they collected a dataset that has a fully connected network. Unfortunately, they released only the network data and the document embeddings but not the raw text. Recently, Li et al. [12] refined this approach.

Another data source that supports abusive language detection is the user's history of previous posts. Qian et al. [20] improved a hate speech classifier for tweets by adding the previous tweets of the author. Raisi and Huang [21] proposed a model that leverages the user's history of posts and the post directed to the user to calculate a bully and victim score for each user. However, to the best of our knowledge, no one has integrated user's previous posts and social networks into abusive language detection.

Besides multimodality, XAI in abusive language detection is another topic that we have to consider in this section. Since XAI is a relatively new field, it has not been frequently applied to abusive language detection with some exceptions [14, 34, 31, 27, 30, 18]. All models use only the text as input, except [30]. Their model also relies on network data. But the network submodel is very simple; it is only a binary vector encoding whether the user follows pre-defined hater accounts. Furthermore, the explanations for this submodel are not detailed. Hence, the explainable model that we propose is an advancement.

3 Data

For our experiment, we use three abusive language datasets that are from Twitter. Table 1 provides an overview of the datasets' characteristics. Figure 1 visualizes the social network graph of the datasets.

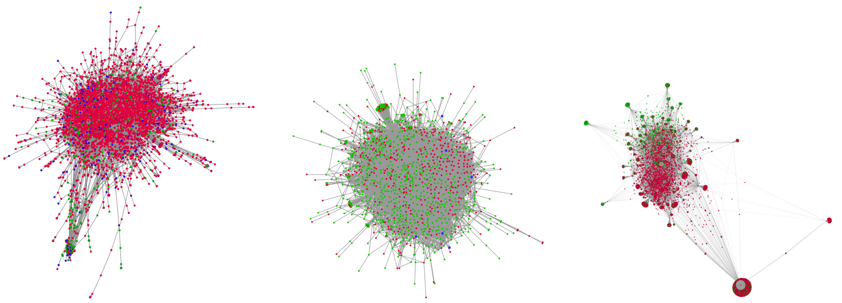
Davidson Davidson et al. [4] released an English abusive language dataset containing 24,783 tweets annotated as hate, offensive, or neither. Unfortunately,

Table 1. Overview of the datasets' statistics

	Davidson			**Waseem**			**Wich**	
Number of tweets	14,939			16,907			68.443	
Number of users	6,725			2,024			939	
Avg. number of tweets per user	2.22			8.35			72.9	
Class	hate	offensive	neither	sexism	racsim	none	offensive	non-offensive
Class distribution	814	11,800	2,325	3,430	1,976	11,501	26,205	42,238
Network: avg. degree	1.85			3.44			1.63	
Network: graph density	0.0005			0.0034			0.0002	

the dataset does not contain any data about the user or the network. Therefore, we used the Twitter API to get the original tweets and the related user and network data. Since not all tweets are still available on Twitter, our dataset has shrunk to 14,939 tweets.

WASEEM Waseem et al. [33] published an English abusive language dataset containing 16,907 tweets annotated as sexist, racist, or none. Similar to DAVIDSON, the dataset does not provide any user- or network-related data. The authors of [15] shared their enriched WASEEM dataset with us containing the user and network data.



(a) DAVIDSON (blue: hateful users, red: offensive users, green: standard user)

(b) WASEEM (blue: racist user, red: sexist user, green: standard user)

(c) WICH (red: offensive user, green: standard user)

Fig. 1. Visual comparison of the network topologies. Standalone nodes or very small subnetworks that do not connect to the main graph for DAVIDSON and WASEEM are excluded.

WICH Wich et al. [35] released a German offensive language dataset containing 4,647,200 tweets annotated as offensive or non-offensive. Most of the tweets are pseudo-labeled with a BERT-based classifier; a smaller portion of the dataset is also manually annotated. The difference between this dataset and the other two is the way it was collected. Wich et al. applied a snowball sampling strategy focusing on users. Starting from seed users, the authors collected the connected users and their tweets based on their offensiveness. Hence, the network graph has a star-shaped network topology contrary to the other two, as depicted in Figure 1c. We select only 68,443 tweets and the related user and network information to better handle the data. The manually annotated tweets are used as a test set.

4 Methodology

The section is split into two subsections. The first one deals with the model architecture and training of the multimodal classification model. The second one considers the XAI technique that we use to explain the predictions of our multimodal model.

4.1 Multimodal Classification Model

Architecture The multimodal classification model for abusive language consists of three submodels that process the different inputs:

1. **Text model**: It processes the text data of the tweet that is meant to be classified. For this purpose, we use DistilBERT with a classification head.
2. **History model**: It processes the tweet history of the user.
3. **Network model**: It processes the social network data of the tweet's user. To model the network data, we use the vector embedding framework GraphSAGE.

The three models' outputs are combined in a linear layer, which outputs the prediction for the tweet to be classified.

Text model The text data of the tweet is fed into a pre-trained DistilBERT model with a classification head. DistilBERT is a lighter and faster version of the transformer-based model BERT [23]. Despite the parameter reduction, its performance is comparable to BERT in general [23] and in the context of abusive language detection [28]. In order to implement the model, we use the Transformers library from Hugging Face[2] and its implementation of DistilBERT [37]. As pre-trained models, we use `distilbert-base-uncased` for the English datasets and `distilbert-base-german-cased` for the German one. Before tokenizing the text data, we remove username mentions from the tweets, but we keep the "@" from the mention[3]. The purpose of this procedure is to avoid the classifier memorizing the username and associating it with one of the classes. But the classifier should recognize that the tweet addresses another user.

[2] https://huggingface.co/transformers/

[3] If a user is mentioned in a tweet, an "@" symbol appears before the user name.

History model We use a bag-of-words model to model the user's tweet history, comprising the 500 most common terms from the dataset based on term frequency-inverse document frequency (tf-idf). For each user, it is a 500-dimensional binary vector that reflects which of the most common terms appear in the user's tweet history.

Network model In order to model the user's social network, we apply the inductive representation learning framework GraphSAGE [9]. The advantage of an inductive learning framework is that it can be applied to previously unseen data, meaning the model can generate an embedding for a new user in a network, which is a desirable property for our use case. Our GraphSAGE model is trained on the undirected network graph of the social relations. Furthermore, we assign to each user/node a class derived from the labels of their tweets. The output of the model is a 32-dimensional graph embedding for each user. The graphs are modeled as follows:

- Davidson: An edge between two users exists if at least one follows the other. A user is labeled as hater, if he or she has at least one hate tweet; as offensive, if he or she has at least one offensive tweet, but no hate tweet; as neither, if he or she has only neither tweets.
- Waseem: An edge between two users exists if at least one follows the other. A user is labeled as racist, if he or she has at least one tweet labeled as racist; same for sexist; as none, if he or she is neither racist nor sexist.
- Wich: An edge between two users exists if at least one has retweeted the other. A user is labeled as offensive, if he or she has at least three offensive tweets.

Users without network connections in their respective dataset, so-called solitary users, do not receive a GraphSAGE embedding; their embedding vector only contains zeros.

The output of the three models is concatenated to a 534 or 535 respectively dimensional vector (DistilBERT: 2 or 3 dimensions depending on the output speech classes; GraphSAGE: 32 dimensions; bag-of-words: 500 dimensions) and fed into a hidden linear layer. This final layer with softmax activation reduces the output to the number of classes according to the selected dataset.

Training Several challenges have to be faced when it comes to training the model. In terms of sampling, we cannot randomly split the dataset: We have to ensure that tweets of any user do not appear in the train and test set; otherwise, we would have a data leakage. Therefore, sampling is done on the user level. Users are categorized into groups based on their class and the existence of a network. We gather six different categories for Waseem and Davidson and four categories for Wich. The train, validation, and test set all contain users from different classes by sampling these categories to prevent bias toward certain user groups. Due to the different tweet counts per user, the train set size varies between 60-70% depending on the dataset.

We under- and oversample the classes during training since all datasets are unbalanced. Moreover, we have to train the three submodels separately because the unsupervised training process of GraphSAGE cannot be combined with the supervised training of DistilBERT. DistilBERT is fine-tuned for two epochs with a batch size of 64 and an Adam optimizer (initial learning rate of 5×10^{-5} and a weight decay of 0.01). We train our GraphSAGE model, consisting of three hidden layers with 32 channels each, for 50 epochs with an Adam optimzer (initial learning rate of 5×10^{-3}). The bag-of-words model does not require training. After training the submodels, we freeze them and train the hidden layer (10 epochs; Adam optimizer with an initial learning rate of 1×10^{-3}).

4.2 Explainable AI Technique

We set model interpretability as a core objective of our work. To this end, we produce Shapley-values-based explanations at different levels of granularity. Shapley values are an established technique to estimate the contribution of input features w.r.t. the model's output [25, 13]. Their suitability for this task has been proven both on a theoretical as well as on an empirical level [13].

As computing exact Shapley values is exponentially complex w.r.t. the input size and hence not feasible, accurate approximations are fundamental for their estimation [13]. As shown in Algorithm 1, we compute them by iteratively averaging each feature's marginal contribution to a specific output class. We find that 15 iterations are sufficient for Shapley values to converge. A random sampling of features was used for reasons of simplicity. Finally, we can assign each feature a Shapley value, representing its relative impact score. A similar approximation approach has been used in [26].

There are two different granularity levels in terms of features: For instance, we can treat each model component (tweet, network, history) as a single feature and derive impact scores (Shapley values) for these components. Alternatively, each model component input or feature (e.g., each token of a tweet) can be treated separately on a more fine-grained level. As Shapley values are additive, they can be aggregated to represent component-level Shapley values. The way feature and components are excluded in order to compute their respective Shapley value changes based on these two levels listed in Table 2. Thus, our multimodal model can be explained on a single instance, and the role played by each model can always be retrieved.

Additionally, we partition the network graph into communities using the Louvain algorithm to derive Shapley values for individual network connections [1]. All user edges in that community with the target user are disabled to obtain the impact of a specific community, resulting in a new GraphSAGE generated user embedding as input for the multimodal model. The embedding vectors of solitary users that only contain zeros result in Shapley values equal to zero for the network component of all these users.

```
Result: Shapley value $\{\phi_t\}_{t=1}^M$ for every feature $\{x_t\}_{t=1}^M$
Input : $p$ sample probability, $x$ instance, $f$ model, $I$ number of iterations
for $i = 0, ..., I$ do
    for $t = 1, ..., M$ do
        sample a Bernoulli vector $P = \{0, 1\}^M$ with probability $p$
        pick $S$ a subset of the features $\{x_t\}_{t=1}^M \setminus \{x_t\}$ according to $P$
        build $x_S$ alteration of $x$ with only features in $S$
        $\phi_t \leftarrow \phi_t \frac{i-1}{i} + \frac{f(x_{S \cup \{x_t\}}) - f(x_S)}{i}$
    end
end
```

Algorithm 1: Shapley value approximation algorithm. In our experiments, $p = 0.7$ and $I = 15$ were used as parameters.

Table 2. Masking strategies for SHAP on component and feature level

	Text	**Network**	**History**
Component wise	Masking BERT output with 0s	Setting GraphSAGE embedding to 0	Setting all vocabulary counts to 0
Feature wise	Masking each token individually	Disabling edges to user based on community and generating new embedding	Setting each vocabulary token count to 0 individually

5 Results

In the first subsection, we deal with answering RQ1 based on the classification performance of our architecture. The second subsection addresses the explainability of the models and related findings to answer RQ2.

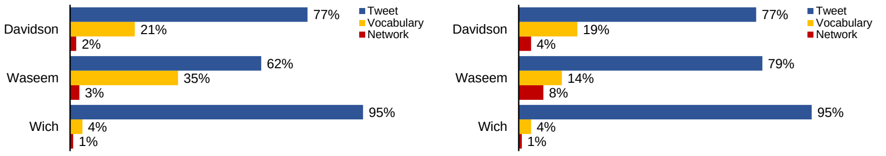
Table 3. Classification models' performance by different architectures and datasets

	Davidson			**Waseem**			**Wich**		
Model	P	R	F1	P	R	F1	P	R	F1
Text	75.3	77.1	76.1	77.5	84.1	80.3	89.8	91.7	90.7
Text + History	73.7	77.8	75.5	79.3	87.8	**82.7**	89.8	91.7	90.7
Text + Network	75.3	77.2	76.2	77.5	84.4	80.4	89.9	91.7	**90.8**
All	74.5	78.9	**76.5**	79.2	88.1	**82.7**	90.0	91.7	**90.8**

5.1 Classification Performance

Table 3 displays the different model architecture performance metrics for the three datasets. We find that combining text, history, and network increases the macro F1 score of Waseem by 2.4 pp and of Davidson by 0.4 pp. In the case of Wich, we observe only a minor increase of the precision by 0.1 pp. We ascribe these diverging increases to two aspects: Firstly, the network of Waseem is the densest one of all three, followed by Davidson and Wich, as depicted in Table 1. Secondly, Wich's text model has a high F1 score, meaning that this submodel presumably drives the predictions of the multimodal model. Our impact analysis using SHAP to identify each submodel's relevance confirms this hypothesis, as depicted in Figure 2. It shows that the network and history data are less relevant for Wich's multimodal model than for the other two models.

In order to answer RQ1, these results signify that leveraging a user's previous posts and their social network data does improve abusive language classification. Additionally, the improvement of the F1 score is proportional to the network's density – the higher the density, the higher the improvement.



(a) Complete test set

(b) Test data that contain network data

Fig. 2. Avg. impact of each classifier's submodels on the respective test set based on Shapley values

5.2 Explainability

In this subsection, we present the results of the XAI technique, SHAP, that we applied to our multimodal model. Firstly, we further investigate the impact of the network and history data added to the text model. Secondly, we show the explanation of a single tweet.

Impact Analysis of the Submodels Figure 2 visualizes the impact of the submodels on the multimodal model. We calculate the impact by aggregating the Shapley values for each submodel based on the tweets in the test set. Figure 2a displays the impact on the complete test set of each dataset, while Figure 2b shows the impact on test data that contains network data[4].

Our first observation is that all classifiers are mainly driven by the text model, followed by the history and network model. Comparing Figure 2a and 2b, we see that network data, if available, contributes to the predictions of Waseem's and Davidson's multimodal models. If we compare the network model's impact of both datasets in the context of network density (Davidson: 5×10^{-4}; Waseem: 3.4×10^{-3}), we can conclude that the denser the network is, the more relevant it is for the classification. These findings confirm our answer to RQ1.

In the case of Waseem, we observe a large contribution of the history model (35%) for the complete test set. We can trace it back to four users that produced a

[4] Network data is not avaiable for all users.

large portion of the dataset and mainly produced all abusive tweets. In general, the number of tweets in the user's history correlates positively with the Shapley value for the history model, reflecting the impact of the history model on the prediction. While the correlation within WICH's dataset is only weak ($r_{Wich} = 0.172$), we observe a moderate correlation for the other two datasets ($r_{Davidson} = 0.500$ and $r_{Waseem} = 0.501$).

Regarding WICH's dataset, the Shapley values indicate that the text model dominates (95%) the multimodal model's prediction, while the other two (4% and 1%) play only a minor role. There are two reasons for this: First, the tweets are pseudo-labeled by a BERT model. Since we use a DistilBERT model similar to BERT, we achieve an outstanding F1 score of the text model (90.7%). The downside of such a good classification performance is that the multimodal model relies mainly on the text model's output. Therefore, the history and network model are less relevant. Furthermore, the dataset's network is characterized by a low degree of interconnectivity compared to the networks of the other two datasets (cf. Table 1).

We established that aggregating the Shapley values of the test set with respect to RQ2 helps us better understand the relevance of each submodel. The insights gained by the applied XAI technique also confirmed our answer to RQ1 that user's network and history data contribute to abusive language detection.

Explaining a Single Tweet Classification After investigating the model on an aggregated level, we focus on explaining the prediction of a single tweet. To do so, we select the following tweet from the DAVIDSON dataset that is labeled and correctly predicted as hateful by our multimodal model:

> *@user i can guarantee a few things: you're white. you've never been anywhere NEAR a real ghetto. you, or a relative is a pig. 100%*

In the following, we demonstrate the explainable capabilities of our multimodal model based on the selected tweet. Figure 3 plots the Shapley values of the tweet's tokens and the user's history and network (last two rows). These Shapley values indicate the relevance of the feature on the multimodal model's prediction as hateful. A positive value (red-colored) represents a contribution favoring the classification as hateful, a negative value (blue-colored) that favors the classification as non-hateful.

We see that the most relevant word for the classification as hateful is "white", which should not be surprising because of the racist context. Furthermore, the @-symbol (representing a user mention) and "you(')re" are relevant for the classification model, indicating that directly addressing someone is recognized as a sign of hate for the classifier. In contrast, the punctuation of the tweet negatively influences the classification as hateful. A possible explanation is that correct spelling and punctuation are often disregarded in the context of abusive language. Beyond the textual perspective, we observe that the history and network submodels favor the classification as hateful. These inputs are relevant for our multimodal model to classify the tweet correctly. Considering Figure 4a (an

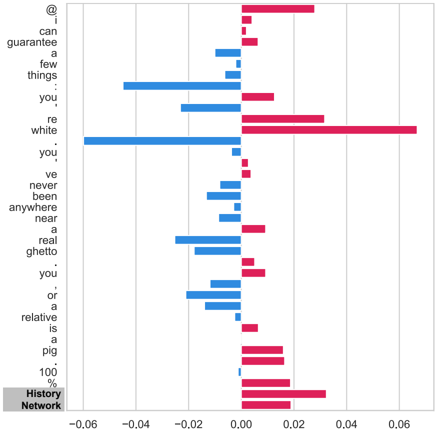
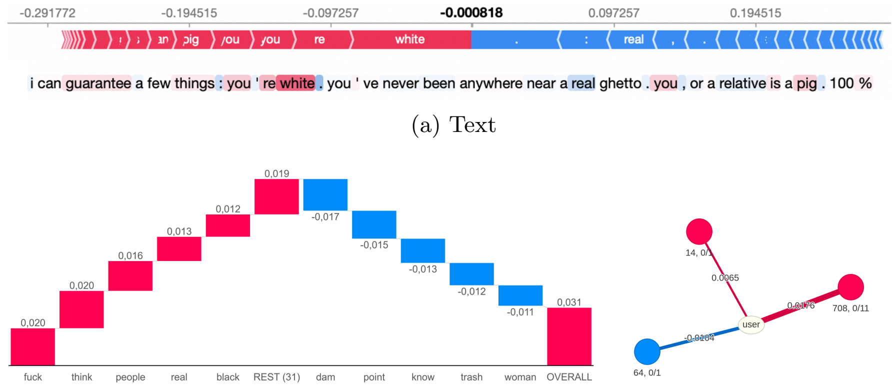


Fig. 3. Relevance of the different features in the form of Shapely values; positive, red values represent favoring a classification as hateful; negative, blue ones the opposite; Shapley values for history and network submodel are aggregated

alternative visualization of the Shapely values), we see that the text model slightly favors the classification as non-hateful, represented by the negative sum of Shapley values. Due to the input from the other two submodel, however, the multimodal model classifies the tweet correctly, making this an excellent example of how abusive language detection can profit from additional data.

Figures 4b and 4c break down the contribution of the history and network model, where Figure 4b is a waterfall chart displaying the most relevant terms that the user used in their previous posts—less relevant terms are summarized in the column named REST. As in the previous charts, red represents a positive contribution to the classification as hateful and blue vice versa. The last column, called OVERALL, is the sum of all terms' Shapley values. In this case, the previous tweets of the user contain words words that are primarily associated with hateful tweets; consequently, the history model favors a classification as hateful. Figure 4c shows the user's ego network and its impact on the classification. The nodes connected to the user represent communities identified by the Louvain algorithm. The first number of a node's label is an identifier; the second number is the number of haters in the community; the third number is the community's total number of users. The color of the nodes and edges have the same meaning as in the other visualizations. In our case, two connected communities contribute to a hateful classification, while the left-pointing community counteracts this.

The presented explanations of the complete model and its submodels provide meaningful and reasonable information to understand better how the model decides to make predictions. These findings extend our answer to RQ2 from the previous section. Our explainable model provides explanations on an aggregated level and a single prediction level to make the classification more understandable.



i can guarantee a few things : you ' re white . you ' ve never been anywhere near a real ghetto . you , or a relative is a pig . 100 %

(a) Text

(b) User's history

(c) User's network (colored nodes represent communities)

Fig. 4. Explanations for predictions of test, history, and network submodel in the form of Shapely values (red, positive values favor a classification as hateful; blue, negative values favor a classification as non-hateful)

6 Discussion

We demonstrated that leveraging a user's history and ego network can improve abusive language detection regarding RQ1, consistent with the findings from other researchers [15, 22, 20]. Our multimodal approach is novel because we combine text, users' previous tweets, and their social relations in one model. The additional data sources provide further indications for the classification model to detect abusive language better. That can be helpful, especially when the classifier struggles with a precise prediction, as in our example in Section 5.2. Other examples are implicit language, irony, or sarcasm, which are hard to detect from a textual perspective. The improvement, however, varies between the datasets. We trace this back to the network density of the available data. Waseem has the network with the highest density and exhibits the best improvement if we integrate history and network data. In contrast, the classification model based on Wich, the dataset with the least dense network, could be improved only slightly. A further difficulty concerning Wich's dataset is that the tweets are pseudo-labeled with a BERT model, and our text submodel uses DistilBERT. Hence, our text submodel performs so well that the multimodal model nearly ignores the outputs of the history and network submodels. Therefore, it was hard to identify any improvement. Relating to Davidson, we had the problem of data degradation. Since the dataset does not contain any user or network data, we used the Twitter API to obtain them. But not all tweets were still available, causing us to use only 60% of the original dataset for our experiment. We require more appropriate datasets to investigate the integration of additional data sources in abusive language detection and refine this approach. For example, Riberio et al. [22] have released a comprehensive dataset containing 4,972 labeled users. Unfortunately, they have not published the tweets of the users. We are aware that releasing a dataset containing social relations and text might violate the users' privacy. Therefore, we suggest anonymizing the data by replacing all user names with anonymous identifiers.

We proved that our multimodal model combined with the SHAP framework provides reasonable and meaningful explanations of its predictions associated with RQ2. These explanations allow us to gain a better understanding with respect of the models in two different ways: (1) the influence of the different submodels on the final predictions on an aggregated level; (2) the relevance of individual features (e.g., word, social relationship) for a single prediction. These explainable capabilities of our multimodal model are a further novelty. To our best knowledge, no one has developed such an explainable model for abusive language detection.

Even though the SHAP explanations are only an approximation, they are necessary for the reliable application of a hate speech detection model, as we have developed. It should be humanly interpretable how each of the three models influences predictions since we combine various data sources, which is especially true when one data source, such as the social network, is not fully transparent for the user. The reason for the missing transparency is that our network submodel learns patterns from social relations, which are more challenging to understand without any additional information than the ones from the text model. Therefore, these explainable capabilities are indispensable for such a system to provide a certain degree of transparency and build trustworthiness.

After focusing on the individual research questions, we have to add an ethical consideration regarding our developed model for various reasons. One may criticize that we integrate social network data, which is personal data, into our model and that the benefit gained by it bears no relation to the invasion of the user's privacy. However, we argue against it based on the following reasons: (1) We use social network data to train embeddings and identify patterns that do not contain any personal data. (2) The user's history and network are shown to enhance the detection rate, even if the used datasets are not the most appropriate ones for this experiment because of the limited density. Furthermore, detecting abusive language can be challenging if the author uses irony, sarcasm, or implicit wording. Therefore, context information (e.g., user's history or network) should be included because its benefit outweighs the damage caused by abusive language.

Another point of criticism could be the possible vulnerability to bias and systematic discrimination of users. In general, DL models are vulnerable to bias due to their black-box nature. In the case of a multimodal model, however, the issue is more aggravated because one submodel can dominate the prediction without any transparency for the user. For example, a model that classifies a user's tweet only because of their social relations discriminates the user with a high probability. We address this challenge by adding explainable capabilities with SHAP. Therefore, we claim that our multimodal model is less vulnerable to bias than classical abusive language detection models applying DL techniques without XAI integration.

7 Conclusion & Outlook

This paper investigated whether users' previous posts and social network data can be leveraged to achieve good, humanly interpretable classification results in the context of abusive language. Concerning the classification performance (RQ1), we showed that the additional data improves the performance depending on the dataset and its network density. For Waseem, we increased the macro F1 score by 2.4 pp, for Davidson by 0.4 pp, and Wich by 0.1 pp. We found that the denser the network, the higher the gain. Nevertheless, the availability of appropriate datasets is a remaining challenge.

The model's interpretability (RQ2) demonstrated that our multimodal model using the SHAP framework produces meaningful and understandable explanations for its predictions. The explanations are provided both on a word level and connections to social communities in the user's ego network. The explanations help better understand a single prediction and the complete model if relevance scores are aggregated on a submodel level. Furthermore, explainability is a necessary feature of such a multimodal model to prevent bias and discrimination.

Integrating a user's previous posts and social network to enhance abusive language detection produced promising results. Therefore, the research community should continue exploring this approach because it might be a feasible way to address the challenge of detecting implicit hate, irony, or sarcasm. Concrete aspects that have to be addressed by future work are the following: (1) collecting appropriate data (in terms of size and network density) to refine our approach, (2) improving our model's architecture.

Acknowledgments

We would like to thank Anika Apel and Mariam Khuchua for their contribution to this project. The research has been partially funded by a scholarship from the Hanns Seidel Foundation financed by the German Federal Ministry of Education and Research.

References

1. Blondel, V.D., Guillaume, J.L., Lambiotte, R., Lefebvre, E.: Fast unfolding of communities in large networks. Journal of Statistical Mechanics: Theory and Experiment **2008**(10), P10008 (Oct 2008)
2. Campbell, W., Baseman, E., Greenfield, K.: Content + context networks for user classification in twitter. In: Frontiers of Network Analysis, NIPS Workshop, 9 December 2013 (2013)

3. Chatzakou, D., Kourtellis, N., Blackburn, J., De Cristofaro, E., Stringhini, G., Vakali, A.: Mean birds: Detecting aggression and bullying on twitter. In: WebSci. pp. 13–22 (2017)
4. Davidson, T., Warmsley, D., Macy, M., Weber, I.: Automated hate speech detection and the problem of offensive language. In: Proc. 11th ICWSM Conf. (2017)
5. Fehn Unsvåg, E., Gambäck, B.: The effects of user features on Twitter hate speech detection. In: Proc. 2nd Workshop on Abusive Language Online (ALW2). pp. 75–85. ACL (2018)
6. Founta, A.M., Chatzakou, D., Kourtellis, N., Blackburn, J., Vakali, A., Leontiadis, I.: A unified deep learning architecture for abuse detection. In: WebSci. pp. 105–114. ACM (2019)
7. Friedman, B., Nissenbaum, H.: Bias in computer systems. ACM Transactions on Information Systems pp. 330–347 (1996)
8. Garland, J., Ghazi-Zahedi, K., Young, J.G., Hébert-Dufresne, L., Galesic, M.: Countering hate on social media: Large scale classification of hate and counter speech. In: Proc. 4th Workshop on Online Abuse and Harms. pp. 102–112 (2020)
9. Hamilton, W.L., Ying, Z., Leskovec, J.: Inductive representation learning on large graphs. In: NIPS. pp. 1024–1034 (2017)
10. Hennig, M., Brandes, U., Pfeffer, J., Mergel, I.: Studying Social Networks. A Guide to Empirical Research. Campus Verlag (2012)
11. Kreißel, P., Ebner, J., Urban, A., Guhl, J.: Hass auf Knopfdruck. Rechtsextreme Trollfabriken und das Ökosystem koordinierter Hasskampagnen im Netz. Institute for Strategic Dialogue (2018)
12. Li, S., Zaidi, N., Liu, Q., Li, G.: Neighbours and kinsmen: Hateful users detection with graph neural network. In: Pacific-Asia Conference on Knowledge Discovery and Data Mining. Springer (2021)
13. Lundberg, S.M., Lee, S.I.: A unified approach to interpreting model predictions. In: NeurIPS (2017)
14. Mathew, B., Saha, P., Yimam, S.M., Biemann, C., Goyal, P., Mukherjee, A.: Hatexplain: A benchmark dataset for explainable hate speech detection. arXiv preprint arXiv:2012.10289 (2020)
15. Mishra, P., Del Tredici, M., Yannakoudakis, H., Shutova, E.: Author profiling for abuse detection. In: COLING. pp. 1088–1098. ACL (2018)
16. Mishra, P., Yannakoudakis, H., Shutova, E.: Tackling online abuse: A survey of automated abuse detection methods. arXiv preprint arXiv:1908.06024 (2019)
17. Molnar, C.: Interpretable Machine Learning (2019), https://christophm.github.io/interpretable-ml-book/
18. Mosca, E., Wich, M., Groh, G.: Understanding and interpreting the impact of user context in hate speech detection. In: Proc. 9th Int. Workshop on Natural Language Processing for Social Media. pp. 91–102. ACL (2021)
19. Papegnies, E., Labatut, V., Dufour, R., Linares, G.: Graph-based features for automatic online abuse detection. In: SLSP. pp. 70–81. Springer (2017)
20. Qian, J., ElSherief, M., Belding, E., Wang, W.Y.: Leveraging intra-user and inter-user representation learning for automated hate speech detection. In: NAACL 2018 (Short Papers). pp. 118–123. ACL (2018)
21. Raisi, E., Huang, B.: Cyberbullying detection with weakly supervised machine learning. pp. 409–416. ASONAM '17, Association for Computing Machinery, New York, NY, USA (2017)
22. Ribeiro, M., Calais, P., Santos, Y., Almeida, V., Meira Jr, W.: Characterizing and detecting hateful users on twitter. In: Proc. Int. AAAI Conf. on Web and Social Media. vol. 12 (2018)

23. Sanh, V., Debut, L., Chaumond, J., Wolf, T.: DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter. In: 2019 5th Workshop on Energy Efficient Machine Learning and Cognitive Computing - NeurIPS 2019 (2019)
24. Schmidt, A., Wiegand, M.: A survey on hate speech detection using natural language processing. In: Proc. 5th Int. Workshop on Natural Language Processing for Social Media. pp. 1–10. ACL (2017)
25. Shapley, L.: Quota solutions of n-person games. Contributions to the Theory of Games **2**, 343–359 (1953)
26. Štrumbelj, E., Kononenko, I.: Explaining prediction models and individual predictions with feature contributions. Knowledge and information systems **41**(3), 647–665 (2014)
27. Švec, A., Pikuliak, M., Šimko, M., Bieliková, M.: Improving moderation of online discussions via interpretable neural models. In: Proc. 2nd Workshop on Abusive Language Online (ALW2). pp. 60–65. ACL (2018)
28. Vidgen, B., Hale, S., Guest, E., Margetts, H., Broniatowski, D., Waseem, Z., Botelho, A., Hall, M., Tromble, R.: Detecting East Asian prejudice on social media. In: Proc. 4th Workshop on Online Abuse and Harms. pp. 162–172. ACL (2020)
29. Vidgen, B., Harris, A., Nguyen, D., Tromble, R., Hale, S., Margetts, H.: Challenges and frontiers in abusive content detection. In: Proc. 3rd Workshop on Abusive Language Online. pp. 80–93. ACL (2019)
30. Vijayaraghavan, P., Larochelle, H., Roy, D.: Interpretable multi-modal hate speech detection. In: Proc. Int. Conf. on Machine Learning AI for Social Good Workshop (2019)
31. Wang, C.: Interpreting neural network hate speech classifiers. In: Proc. 2nd Workshop on Abusive Language Online (ALW2). pp. 86–92. ACL (2018)
32. Waseem, Z., Davidson, T., Warmsley, D., Weber, I.: Understanding abuse: A typology of abusive language detection subtasks. In: Proc. 1st Workshop on Abusive Language Online. pp. 78–84. ACL (2017)
33. Waseem, Z., Hovy, D.: Hateful symbols or hateful people? predictive features for hate speech detection on Twitter. In: Proc. NAACL Student Research Workshop. pp. 88–93. ACL (2016)
34. Wich, M., Bauer, J., Groh, G.: Impact of politically biased data on hate speech classification. In: Proc. 4th Workshop on Online Abuse and Harms. pp. 54–64. ACL (2020)
35. Wich, M., Breitinger, M., Strathern, W., Naimarevic, M., Groh, G., Pfeffer, J.: Are your friends also haters? identification of hater networks on social media: Data paper. In: Companion Proc. Web Conference 2021. ACM (2021)
36. Williams, M.L., Burnap, P., Javed, A., Liu, H., Ozalp, S.: Hate in the machine: Anti-black and anti-muslim social media posts as predictors of offline racially and religiously aggravated crime. The British Journal of Criminology pp. 93–117 (2020)
37. Wolf, T., Debut, L., Sanh, V., Chaumond, J., Delangue, C., Moi, A., Cistac, P., Rault, T., Louf, R., Funtowicz, M., Davison, J., Shleifer, S., von Platen, P., Ma, C., Jernite, Y., Plu, J., Xu, C., Le Scao, T., Gugger, S., Drame, M., Lhoest, Q., Rush, A.: Transformers: State-of-the-art natural language processing. In: Proc. 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations. pp. 38–45. ACL (2020)
38. Xu, J., Lu, T.C., et al.: Automated classification of extremist twitter accounts using content-based and network-based features. In: 2016 IEEE Int. Conf. on Big Data. pp. 2545–2549. IEEE (2016)

B.3 STUDY VI

Lukas Huber, Marc Alexander Kühn, Edoardo Mosca, and Georg Groh (May 2022). "Detecting Word-Level Adversarial Text Attacks via SHapley Additive exPlanations." In: *Proceedings of the 7th Workshop on Representation Learning for NLP*. Dublin, Ireland: Association for Computational Linguistics, pp. 156–166. DOI: 10.18653/v1/2022.repl4nlp-1.16. URL: https://aclanthology.org/2022.repl4nlp-1.16

[2] https://creativecommons.org/licenses/by/4.0/

Publication Summary

"State-of-the-art machine learning models are prone to adversarial attacks: Maliciously crafted inputs to fool the model into making a wrong prediction, often with high confidence. While defense strategies have been extensively explored in the computer vision domain, research in natural language processing still lacks techniques to make models resilient to adversarial text inputs. We adapt a technique from computer vision to detect word-level attacks targeting text classifiers. This method relies on training an adversarial detector leveraging Shapley additive explanations and outperforms the current state-of-the-art on two benchmarks. Furthermore, we prove the detector requires only a low amount of training samples and, in some cases, generalizes to different datasets without needing to retrain." (Huber et al., 2022, p. 1)

Author Contributions

Edoardo Mosca contributed to the study as follows:

- Conception, development, and lead of the research project **100%**
- Literature review and feasibility study **50%**
- Setup and Implementation of experiments **20%**
- Analysis and interpretation of results. **20%**
- Drafting of the manuscript **50%**
- Submission, peer review, and publication process **80%**

Detecting Word-Level Adversarial Text Attacks via SHapley Additive exPlanations

Lukas Huber[*]
TU Munich,
Department of Informatics,
Germany
lukas1.huber@tum.de

Marc Alexander Kühn[*]
TU Munich,
Department of Informatics,
Germany
marcalexander.kuehn@tum.de

Edoardo Mosca[*]
TU Munich,
Department of Informatics,
Germany
edoardo.mosca@tum.de

Georg Groh
TU Munich,
Department of Informatics,
Germany
grohg@in.tum.de

Abstract

State-of-the-art machine learning models are prone to adversarial attacks: Maliciously crafted inputs to fool the model into making a wrong prediction, often with high confidence. While defense strategies have been extensively explored in the computer vision domain, research in natural language processing still lacks techniques to make models resilient to adversarial text inputs. We adapt a technique from computer vision to detect word-level attacks targeting text classifiers. This method relies on training an adversarial detector leveraging Shapley additive explanations and outperforms the current state-of-the-art on two benchmarks. Furthermore, we prove the detector requires only a low amount of training samples and, in some cases, generalizes to different datasets without needing to retrain.

1 Introduction

Adversarial examples are slightly perturbed input samples purposely crafted to fool a target model (Szegedy et al., 2014). Despite being similar to the original samples, they are often misclassified with high confidence (Goodfellow et al., 2015). Without effective defense techniques, machine learning models become unusable in high-stakes situations and safety-critical tasks (Sharma et al., 2019).

Research in computer vision has extensively worked on better understanding adversarial image attacks and developing more robust models (Madry et al., 2018; Ozdag, 2018). However, the literature in *Natural Language Processing (NLP)* has witnessed fewer advances concerning this issue (Mozes et al., 2021; Zhou et al., 2019; Wang et al., 2019).

Text data needs to fulfill several properties such as lexical, grammatical, and semantic constraints. Thus, many efficient adversarial image attacks—e.g. gradient-based ones—are not transferable as they would lead to incorrect characters and non-existing terms (Zhang et al., 2020). However, word-level attacks that can preserve semantical information without introducing noticeable inconsistencies are particularly effective and not detectable via spell checkers (Garg and Ramakrishnan, 2020; Ren et al., 2019).

The lack of defense strategies against word-level text attacks motivates our research as this is a major obstacle to the safe deployment of NLP models. This work's contribution can be summarized as follows:

(1) Based on an analogous idea from computer vision (Fidel et al., 2020), we propose an adversarial attack detector leveraging *SHapley Additive exPlanations* (SHAP) to accurately recognize input manipulations (Lundberg and Lee, 2017). Results show that it outperforms the previous state of the art in adversarial detection on multiple datasets (Mozes et al., 2021).

(2) We analyze our method in terms of data efficiency and generalization. The proposed approach still offers competitive performance when trained on very little data and can even be transferred to unseen datasets while almost matching the previous state of the art.

[*] These authors contributed equally

Proceedings of the 7th Workshop on Representation Learning for NLP, pages 156 - 166
May 26, 2022 ©2022 Association for Computational Linguistics

(3) Alongside the quantitative analysis and its results, we visualize the space of generated Shapley-value-based explanations. This qualitative analysis sheds light on the reasons behind our method's high performance and desirable properties.

2 Related Work

2.1 Adversarial Text Attacks

An adversarial text attack is an artificial input obtained by modifying a sample from the available data. Normally, the altered text is similar—syntactically, semantically, or both—to the original one. However, their corresponding classification output substantially differs. Attacks can be either *targeted* or *untargeted* (Tao et al., 2018). Attacks of the first type aim to create misclassification results w.r.t. a specific class whereas the latter type wants to generate a misclassification regardless of the exact class.

Methods like DeepWordBug (Gao et al., 2018) or Hotflip (Ebrahimi et al., 2018) introduce character-level noise to create typos and grammatical inconsistencies in the sentence. These adversarial examples appear very similar to the original samples, but do not perfectly preserve their meaning and can be recognized due to their lexical incorrectness.

Other types of attacks instead alter the text at the word level and produce semantically equivalent and grammatically correct sentences to the initial input. Examples of techniques using this strategy are PWWS (Ren et al., 2019), TextFooler (Jin et al., 2020), and BAE (Garg and Ramakrishnan, 2020).

2.2 Defense Strategies for Computer Vision

Robustness against adversarial attacks—and especially their automatic detection—has been more exhaustively researched for computer vision applications rather than for text inputs. Hence, we briefly present a selection of the most promising approaches.

Xu et al. (2018) propose *Feature Squeezing*, based on the assumption that feature spaces are often unnecessarily large and leave extensive possibilities for an attacker to generate adversarial examples. Their approach leverages this fact by comparing the prediction of the original input image with a simplified one. When this difference surpasses a specific threshold, the input is classified as adversarial.

Roth et al. (2019) detect adversarial examples by measuring statistical differences between original and perturbed logits. According to their results, output logits corresponding to adversarial examples exhibit a much larger variation than normal samples when the input is perturbed.

Integrating explainability to detect adversarial examples has already been shown to be beneficial. Fidel et al. (2020) detect patterns in the SHAP signatures of input images (Lundberg and Lee, 2017). For normal samples, the inter-class SHAP signatures share common characteristics. For adversarial examples, however, the SHAP signatures show a mixture between two classes which can easily be detected using an additional classification model.

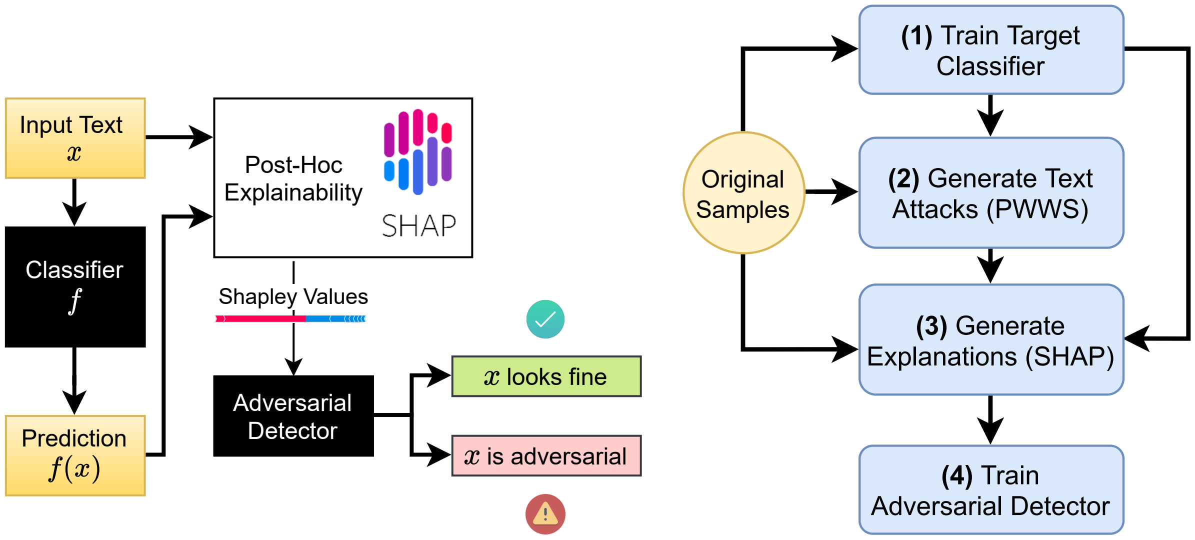
2.3 Defense Strategies for Natural Language Processing

Character-level attacks can be countered with defenses based on spell checkers (Pruthi et al., 2019; Huang et al., 2019). Nonetheless, those same defenses are extremely vulnerable to word-level attacks capable of preserving language coherence (Wang et al., 2019). Effective methods against syntactically correct attacks are *Adversarial Training* (AT) (Goodfellow et al., 2015), *Dirichlet Neighborhood Ensemble* (DNE) (Zhou et al., 2020), *Adversarial Sparse Convex Combination* (ASCC) (Dong et al., 2021) and *Synonym Encoding Method* (SEM) (Wang et al., 2019). The first three leverage some form of data augmentation to train the model on perturbed samples as well. The last, instead, introduces an encoder step before the target model's input layer and trains it to eliminate potential perturbations.

Particularly relevant for this work are *adversarial detection* methods. In contrast to other defenses, they can explicitly recognize manipulated inputs and send an alert signal. For natural language data, the available methods are *Frequency-Guided Word Substitution* (FGWS) (Mozes et al., 2021) and *learning to DIScriminate Perturbation* (DISP) (Zhou et al., 2019). The first—exploiting frequency properties of adversarial words—is the most recent and accurate method. Its authors showed medium to high F1 detection scores in a range from 62.2-91.4%, varying on the type of attack and target model.

2.4 Feature Relevance Explainability Methods

Among explainability techniques, *feature relevance* methods are often used to explain predictions pro-



(a) Goal Pipeline

(b) Construction Steps

Figure 1: Our detector for recognizing adversarial examples: the overall pipeline once the detector is trained (a) and the necessary steps in order to train it (b). While generating many adversarial attacks and explanations is required for training, the detector can then be simply "plugged in" and deployed together with the classifier f.

duced by black-box models (Arrieta et al., 2020; Mosca et al., 2021). Their goal is to attribute a relevance score to each input feature. Such value should quantify the effect that the feature has on the output, i.e. their contribution to the model's prediction (Wich et al., 2021).

Some of these methods rely on computing the gradient of the output w.r.t. the input features (Simonyan et al., 2014; Sundararajan et al., 2017). Others, such as LRP (Bach et al., 2015) and DeepLIFT (Shrikumar et al., 2017), are specifically designed for neural networks and follow the information flow in a backward fashion through the model's architecture. The procedure continues one layer at a time until the input features are reached. LIME (Ribeiro et al., 2016) explains black-box models via a local surrogate that approximates their behavior around a single instance. The surrogate can be then interpreted directly to estimate each feature's relevance.

Lundberg and Lee (2017) prove that several popular feature relevance methods—including LIME, LRP, and DeepLIFT—belong to a broader class of approaches: *additive feature relevance methods*. The authors propose a unified view of such methods that, combined with the game-theoretic concept of Shapley values (Shapley, 1952), constitutes the SHAP framework. SHAP-based explanations are covered more in detail in Section 3.2 as they represent a fundamental component of our proposed method.

3 Methodology

Our defense belongs to the adversarial detection category and is strongly inspired by the work of Fidel et al. (2020), which detects image-based adversarial attacks for computer vision models by using SHAP signatures. This work, instead, studies the application of this idea to text attacks for NLP classifiers. As sketched in Figure 1a, our goal pipeline consists of multiple stages. First, the input is fed to a classifier trained on the task-at-hand, which outputs a prediction. Shapley values are then computed w.r.t. the outcome and passed onto a machine-learning detector that predicts whether the sample is an adversarial attack. Note that our detector does not make any assumption on the classifier and is hence model-agnostic.

The classifier targeted by the attacks becomes considerably more robust when used in combination with the adversarial detector. To achieve our goal, we have to take several steps in order to train our detector. These steps—also summarized in Figure 1b for the reader—are described in detail in the next sections.

3.1 Crafting Adversarial Text Attacks

To train and test our detector, we choose to craft attacks semantically similar to the original input. This choice preserves lexical and grammatical co-

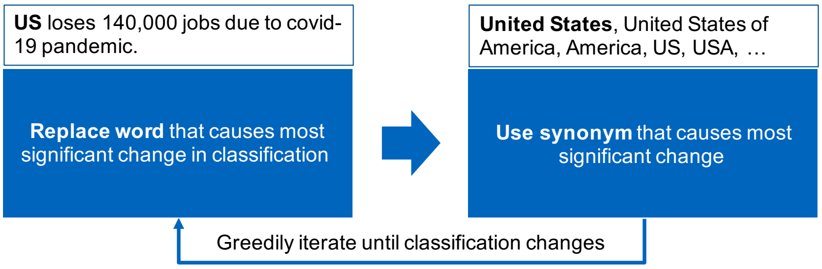

US loses 140,000 jobs due to covid-19 pandemic.
Replace word that causes most significant change in classification
United States, United States of America, America, US, USA, …
Use synonym that causes most significant change
Greedily iterate until classification changes

Figure 2: A simplified view of the generation of adversarial examples using PWWS (Ren et al., 2019)

herence also in adversarial sentences. We believe that such attacks are more subtle as they cannot be detected by spell checkers. In practice, for each sample x in the dataset, we generate

$$x^* = x + \Delta x, \|\Delta x\| < \epsilon \tag{1}$$

where Δx is a semantic perturbation and the classes predicted for x and x^* are different. To this end, we utilize the untargeted *Probability Weighted Word Saliency* (PWWS) method by Ren et al. (2019). This approach shows high effectiveness with good transferability. According to human evaluation, PWWS provides realistic examples with lexical correctness and only sporadic grammatical errors or semantic shifting (Ren et al., 2019).

The technique selects the word to be replaced based on two factors. The first is the change in the classification probability after substitution. The second, called *word saliency*, measures the variation in the output probability of the classifier if the word is set to unknown (out of vocabulary). The chosen word is then replaced by a word from a synonym set which causes the most significant change of classification probability. The algorithm greedily iterates until enough words have been replaced to change the final classification label. Figure 2 sketches the core idea behind the method.

3.2 Generating Model Explanations

Whenever classifying an input sentence as either regular or adversarial, our detector needs access to its corresponding feature relevance explanation. In other words, the detector takes its decision based on *how strong* each feature—in our case each word—influences the final model prediction. The assumption is that the model's reaction to original and adversarial samples is different even if the inputs look similar for a human. Thus, the model explanations for the two samples should also substantially differ from each other (Fidel et al., 2020).

We pick SHAP (Lundberg and Lee, 2017) to produce instance-level explanations to train the adversarial detector. This choice is motivated by the empirical superiority proven by its developers (Lundberg and Lee, 2017) and its previous successful applications in detecting attacks in computer vision. However, while Fidel et al. (2020) generate SHAP signatures w.r.t. the penultimate layer of the target model, we produce explanations directly w.r.t. the input sentence as text perturbations are introduced at the word level.

SHAP is based on a game theory concept—called Shapley values (Shapley, 1952)—originally used to fairly distribute a reward to a set of players that contributed to a certain outcome. In our case, the outcome is the model's prediction whereas the input features, i.e. the input words, are the players involved. Since the players most likely contributed differently to the turnout, their payout should differ based on their impact. Given a text classifier f and the set of all available features M, the Shapley value corresponding to each feature i is computed independently. More precisely, it is a weighted average of the relative outcome differences

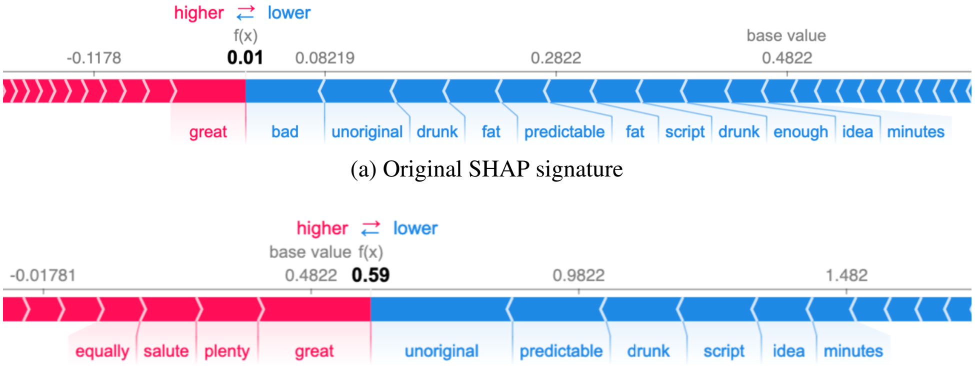
$$f(S \cup \{i\}) - f(S) \tag{2}$$

across all feature subsets $S \subseteq M \setminus \{i\}$.

As there are $2^{|M|}$ possible choices for S, exact Shapley values are exponentially complex to compute. However, the SHAP framework offers several methods to approximate them accurately and efficiently (Lundberg and Lee, 2017). In our work, we utilize DeepSHAP as it is tailored to deep learning models, which we utilize as targets for the text attacks (Lundberg and Lee, 2017). An official implementation has been made publicly available by the SHAP authors. [1]

Figure 3 shows two examples of explanations generated for *IMDb*, a movie review dataset (Maas

[1] https://github.com/slundberg/shap



(a) Original SHAP signature

(b) Adversarial SHAP signature

Figure 3: Force plots generated for a sample of the *IMDb* dataset and its corresponding adversarial attack. The *base value* indicates the average model's prediction across the whole dataset and $f(x)$ represents the model output probability for the selected instance. Red attributes drive the predictions towards class 1 (i.e. a positive review) and blue ones towards class 0 (i.e. a negative review). Starting from the base value (~ 0.48) and adding up all word contributions we reach the final prediction of 0.01. Hence, the original sample is classified as negative with high confidence. In the adversarial SHAP signature, most negative words were replaced by synonyms such that the prediction is now positive.

et al., 2011), with DeepSHAP. The first (Figure 3a) was generated from an original sample while the second (Figure 3b) from its corresponding adversarial attack generated with PWWS. As we can see, the attack changes substantially the effect that words have on the prediction. Hence, word-level contributions are a major indicator for detecting parts of a sentence that have a suspiciously high impact on the model decision. This supports our initial hypothesis that SHAP explanations do not rely on image-only properties and therefore can also serve as features for an adversarial detector in the NLP domain.

3.3 Target Model and Detector Architectures

Our pipeline includes two machine learning models: the text classifier trained for the task-at-hand and the adversarial detector.

For consistency with Mozes et al. (2021), used later for performance comparison, we chose a Bidirectional LSTM (Bi-LSTM) (Schuster and Paliwal, 1997) as architecture to be targeted by the adversarial attacks. However, other NLP models can also be utilized as the detector does not make any assumption on the classifier. The text inputs are first trimmed and padded to an equal length of 100. Increasing the input length drastically increases complexity along the pipeline while only yielding minor accuracy gains. Tokens are transformed into GloVe embeddings (Pennington et al., 2014) before being fed to the Bi-LSTM core layer. We attach a fully connected head layer to compute output probabilities. We adjust the number of output neurons based on the dataset currently in use.

SHAP values are extracted from the model for all output classes. Therefore, the SHAP signatures passed to the detector are numerical vectors of dimensionality [#classes × 100]. Here, each numerical value corresponds to the impact of a single word w.r.t. the model's output. We do not pick any particular architecture for our adversarial detector. Instead, we experiment with a variety of relatively simple machine learning models to test their performance. We include a *random forest* (Breiman, 2001), a *Support Vector Machine* (SVM) (Boser et al., 1992), and a simple two-layer-feed-forward neural network (Rumelhart et al., 1985).

3.4 Overall Pipeline and Experimental Setup

With the methodology for the main steps outlined in the previous sections, we now describe in greater detail how those steps are combined, following what we initially presented in Figure 1b. We repeat the procedure for each text dataset utilized for testing. These will be presented later in our evaluation section (4).

To begin with, we train the Bi-LSTM model on the given dataset. We consider this step concluded once the model converges to a satisfactory accuracy. This is usually around 90% accuracy, depending on the dataset. After that, we utilize PWWS as proposed by Ren et al. (2019)—implemented

	Method	AG_News	IMDb	SST-2	Yelp Polarity	Metric
Our	Neural Network	0.90 / 0.90	**0.96 / 0.96**	0.75 / 0.75	**0.94 / 0.94**	F1 score / Accuracy
	Random Forest	**0.91 / 0.91**	0.87 / 0.87	**0.77 / 0.77**	0.84 / 0.84	F1 score / Accuracy
	SVM	0.90 / 0.90	0.90 / 0.90	0.74 / 0.74	0.89 / 0.89	F1 score / Accuracy
SotA Detector	FGWS (Mozes et al., 2021)	-	0.77	0.63	-	F1 score
Other Defenses	DNE (Zhou et al., 2020)	**0.91**	0.82	-	-	Accuracy
	SEM (Wang et al., 2019)	0.76	0.85	-	-	Accuracy
	ASCC (Dong et al., 2021)	-	0.77	-	-	Accuracy

Table 1: Performance of different detector architectures on the *AG_News, IMDb, SST-2* and *Yelp Polarity* datasets. For comparison, we report also the defense performance of *Frequency-Guided Word Substitutions* (FGWS), *Dirichlet Neighbourhood Ensemble* (DNE), *Synonym Encoding Method* (SEM) and *Adversarial Sparse Convex Combinations* (ASCC).

in the TextAttack library [2]—to produce adversarial attacks targeting our trained NLP model. We generate one attack for each sample in the dataset. Instance-level explanations—i.e. Shapley value approximations—are then created via SHAP, both for normal and adversarial samples (Lundberg and Lee, 2017).

We combine all explanations to compose a balanced dataset for our adversarial detector. The data is split into training and test sets following an 80/20-ratio. We further used the default hyperparameters for all models in the framework. To allow for optimal reproducibility, we seeded all of our experiments. For the neural network-based detector, we pick layers of size 400 using a ReLU activation and an L1 weight regularizer to avoid overfitting. To further increase regularization, Dropout is used (Srivastava et al., 2014). The model is then trained for 10 epochs using the Adam optimizer with a learning rate of 0.001 and β_1, β_2 set to their default values of 0.9 and 0.99 respectively (Kingma and Ba, 2015).

4 Evaluation

4.1 Performance Results

We evaluate our approach on four major datasets often used in research, namely *IMDb* (Maas et al., 2011), *SST-2* (Socher et al., 2013), *Yelp Polarity* and *AG_News* (Zhang et al., 2015). While the last one classifies news articles into four distinct categories, the other three are binary sentiment analysis tasks on movie review data. The reviews are not fed into the detector directly but their corresponding SHAP signatures are instead. The number of samples in the datasets used for the experiment is reported in Table 2. Every dataset consists of a 50:50 split between original and adversarial samples and the sizes are varying between 940 (*Yelp Polarity*) and 100,000 (*AG_News*) samples.

Dataset	Size	#Normal	#Adversarial
AG_News	100,000	50,000	50,000
IMDb	3,580	1,790	1,790
SST-2	3,162	1,581	1,581
Yelp Polarity	940	470	470

Table 2: Sizes of the individual SHAP signature datasets used for training the adversarial detector. All datasets consist of 50% normal and 50% adversarial signatures.

Table 1 shows the performance of various detector architectures on the four datasets together alongside results achieved by previously proposed methods. To the best of our knowledge, the FGWS method proposed by Mozes et al. (2021) is the best detector currently available. With our SHAP-based classifiers, we significantly outperform their method on the *IMDb* dataset by 19% with an F1-score of 96% and on the *SST-2* dataset by 14% with an F1-score of 77%. Relatively simple machine learning models like a random forest or a support vector machine are able to classify the data very accurately. Both Mozes et al. (2021) and our work evaluate their defenses against PWWS targeting a Bi-LSTM model.

Besides adversarial detectors, we also outperform all other existing defenses to the best of our knowledge. On *IMDb*, our approach improves by 11% accuracy compared to the best method (Wang et al., 2019). On *AG_News*, it is matched only by the DNE method from Zhou et al. (2020). For each approach considered, we report the result w.r.t. the configuration achieving the best performance against PWWS from their corresponding original work. For completeness, we mention that Zhou et al. (2019) reports great results but their performance is not comparable as they do not test their method against any well-established attack.

[2] https://github.com/QData/TextAttack

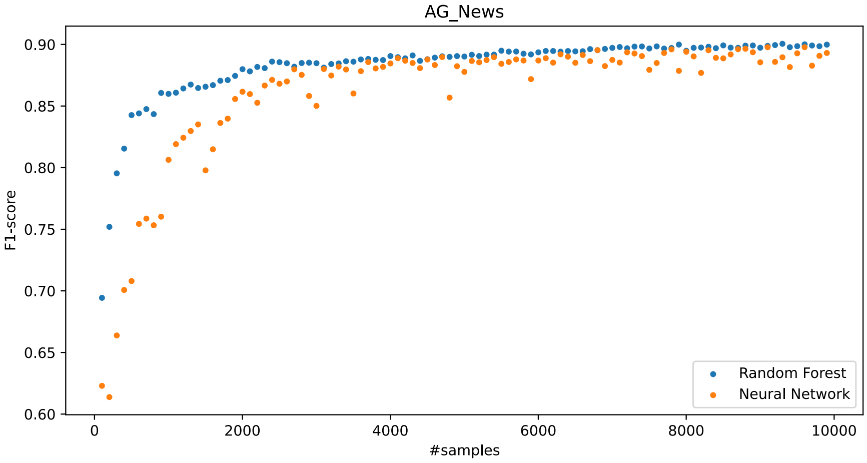


Figure 4: F1-scores for independent runs on the *AG_News* dataset using differently sized subsets of the training data. The F1-score starts to plateau after a few thousand samples for all detectors which shows data efficiency.

Classifier	Unnormalized SHAP	Unnorm. SHAP + Predicted Class	Normalized SHAP
Neural Network	0.90	0.90	0.90
Random Forest	0.91	0.91	0.92
SVM	0.90	0.90	0.90
Linear SVM	0.67	0.67	0.65

Table 3: F1-scores of input modifications for the detectors on the *AG_News* dataset.

To further improve the predictive performance of the model, we also included the predicted class coming from the base model as an input feature for the detector. As shown in Table 3, this had neither a positive nor a negative influence on the performance of the model. Normalizing the SHAP signatures only led to minor improvements for random forests and neural networks. This can be explained by the fact that all input features are Shapley values and are therefore in the same range.

4.2 Transferability

Base-Model	IMDb (Test)	SST-2 (Test)
IMDb	-	0.56
SST-2	0.42	-
Yelp Polarity	**0.71**	**0.66**

Table 4: F1-scores of the inference step with *IMDb* and *SST-2* datasets on neural network base-models which were trained on *IMDb, SST-2* and *Yelp Polarity*.

During our research the question arose whether the detectors are agnostic to the dataset or highly specialized. To evaluate this property, we trained three base-models with a neural network backbone on the *IMDb*, *SST-2* and *Yelp Polarity* datasets. Then, we performed the inference step with the *IMDb* and *SST-2* test sets on all three detectors and observed how the performance varies with different dataset combinations.

The results can be seen in Table 4. We report the strongest results when the detector was tested on the same dataset that was also used during training. This resulted in our competitive F1-scores of 94% on *IMDb* and 77% on *SST-2*. Interestingly, there existed other combinations which also produced results comparable to the state of the art, although the performance dropped compared to our strongest detectors. To be precise, the base-model which was trained on *Yelp Polarity* achieved good F1-scores on test sets of *IMDb* with 71.5% and of *SST-2* with 66%. In comparison, the state-of-the-art detector tested with similarly generated adversarial samples on a LSTM with PWWS by Mozes et al. (2021) achieved F1-scores of 77.4% on *IMDb* and of 63.4% on *SST-2*.

Such results are yet not strong enough to prove full generalization capabilities. However, we find them promising as they indicate that our detectors are in some cases actually transferable to other datasets once trained. Future research is crucial as in practice it allows to reuse models for different tasks.

4.3 Data efficiency

While our approach offers state-of-the-art detection performance of adversarial attacks, the corresponding detector model can be trained with a surprisingly low amount of data. To evaluate this property, we trained a neural network and a random forest on incremental subsets of the *IMDb* dataset where all runs were conducted independently from each other. We started with a dataset size of 100 and incrementally increased the number of samples up to 10,000. From Figure 4 one can directly observe the limited amount of data needed for the model to converge. For a neural network about 4,000 samples are needed before the F1-score starts to plateau. For a random forest classifier even less data is sufficient with around 3,000 samples.

4.4 Qualitative Results

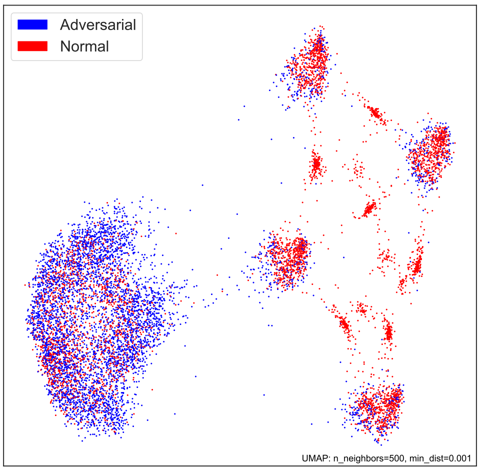


Figure 5: Visualization of the SHAP signatures of the *AG_News* dataset using UMAP. We randomly selected 10% of the samples to avoid overplotting.

In order to understand how the detector is able to distinguish between normal and adversarial inputs, we visualized the SHAP signatures in a two-dimensional space. To project the samples we rely on the UMAP dimensionality reduction algorithm proposed by McInnes et al. (2020). It is based on the fact that most high-dimensional data actually lies on a much lower-dimensional manifold and can be explained by a reduced number of variables. Figure 5 clearly shows four distinct red clusters corresponding to the four classes of the *AG_News* dataset. Regardless of their original class, most of the adversarial samples collapse into a single cluster which is clearly separable from the others. This explains why rather simple detector models are sufficient to accurately differentiate between normal and adversarial inputs. Our result is consistent with the experiments done by Fidel et al. (2020) which performed a similar analysis on SHAP signatures for images from the CIFAR-10 dataset (Krizhevsky et al., 2009).

4.5 Limitations

After the success in computer vision (Fidel et al., 2020), this work shows that SHAP values are also a valuable asset for discriminating between original and adversarial text samples. However, while word-level explanations are particularly effective at detecting word-level attacks, it is unclear how they would transfer to more sophisticated text manipulations. We believe this is a vulnerability as future attacks could involve using negations or paraphrasing whole sentences instead of unigrams.

While the approach's pipeline is intuitive and the results look promising, further research needs to study transferability to more complex target models such as transformers architectures. At the same time, we hope that future research also focuses on creating standard benchmarks to facilitate performance comparisons with previous defense methods.

5 Conclusion

Adversarial text examples are a major challenge for current research and represent an obstacle for safely deploying NLP models in high-stakes applications. While attacks are hard to be distinguished from their corresponding originals, patterns in the model's reaction can be recognized and leveraged using SHAP signatures for detecting manipulated input samples.

Our work trains a machine learning detector using SHAP explanations of normal and adversarial samples generated with PWWS. The proposed method is both intuitive and effective since it allows to detect parts of a sentence that have a suspiciously high impact on the model prediction and therefore distinguishes between regular and manipulated samples. Furthermore, our detector is model-agnostic as it does not make any assumption on the classifier targeted by the attacks.

Our approach achieves high accuracy and considerably outperforms the previous state of the art. In terms of data efficiency, we prove that the method can achieve nearly optimal performance also when using a small portion of the available data for training. A qualitative analysis of the SHAP signature landscape shows most adversarial samples contained in a single cluster, suggesting that model explanations explicitly encode information to separate attacks from their counterpart. We believe this result explains why relatively simple detector architectures suffice to achieve good performance results.

In terms of transferability to multiple datasets, our results are promising but yet not sufficient to prove full generalization capabilities. Although in some cases we match state-of-the-art performance even when training on one dataset and testing on another, our results are highly dependent on the dataset pair.

We encourage future research to continue working on generalization across multiple data sources and to evaluate performance against multiple types of attacks and models. We believe our contribution can help researchers to develop better defense strategies against attacks and thus promoting the safe deployment of NLP models in practice. We release our code to the public to facilitate further research and development [3].

References

Alejandro Barredo Arrieta, Natalia Díaz-Rodríguez, Javier Del Ser, Adrien Bennetot, Siham Tabik, Alberto Barbado, Salvador García, Sergio Gil-López, Daniel Molina, Richard Benjamins, et al. 2020. Explainable artificial intelligence (xai): Concepts, taxonomies, opportunities and challenges toward responsible ai. *Information Fusion*, 58:82–115.

Sebastian Bach, Alexander Binder, Grégoire Montavon, Frederick Klauschen, Klaus-Robert Müller, and Wojciech Samek. 2015. On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation. *PloS one*, 10(7):130–140.

Bernhard E Boser, Isabelle M Guyon, and Vladimir N Vapnik. 1992. A training algorithm for optimal margin classifiers. In *Proceedings of the 5th Annual ACM Workshop on Computational Learning Theory*, pages 144–152.

Leo Breiman. 2001. Random forests. *Machine learning*, 45(1):5–32.

Xinshuai Dong, Anh Tuan Luu, Rongrong Ji, and Hong Liu. 2021. Towards robustness against natural language word substitutions. In *9th International Conference on Learning Representations (ICLR)*.

Javid Ebrahimi, Anyi Rao, Daniel Lowd, and Dejing Dou. 2018. 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.

Gil Fidel, Ron Bitton, and Asaf Shabtai. 2020. When explainability meets adversarial learning: Detecting adversarial examples using SHAP signatures. In *2020 International Joint Conference on Neural Networks (IJCNN)*. IEEE.

Ji Gao, Jack Lanchantin, Mary Lou Soffa, and Yanjun Qi. 2018. Black-box generation of adversarial text sequences to evade deep learning classifiers. In *2018 IEEE Security and Privacy Workshops (SPW)*, pages 50–56. IEEE.

Siddhant Garg and Goutham Ramakrishnan. 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.

Ian J. Goodfellow, Jonathon Shlens, and Christian Szegedy. 2015. Explaining and harnessing adversarial examples.

Po-Sen Huang, Robert Stanforth, Johannes Welbl, Chris Dyer, Dani Yogatama, Sven Gowal, Krishnamurthy Dvijotham, and Pushmeet Kohli. 2019. Achieving verified robustness to symbol substitutions via interval bound propagation. 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 4083–4093.

Di Jin, Zhijing Jin, Joey Tianyi Zhou, and Peter Szolovits. 2020. Is bert really robust? a strong baseline for natural language attack on text classification and entailment. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pages 8018–8025.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

Alex Krizhevsky et al. 2009. Learning multiple layers of features from tiny images.

Scott M. Lundberg and Su-In Lee. 2017. A unified approach to interpreting model predictions. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, page 4768–4777, Red Hook, NY, USA. Curran Associates Inc.

Andrew L. Maas, Raymond E. Daly, Peter T. Pham, Dan Huang, Andrew Y. Ng, and Christopher Potts. 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.

[3] `https://github.com/huberl/adversarial_shap_detect_Repl4NLP`

Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. 2018. Towards deep learning models resistant to adversarial attacks. In *International Conference on Learning Representations*.

Leland McInnes, John Healy, and James Melville. 2020. Umap: Uniform manifold approximation and projection for dimension reduction.

Edoardo Mosca, Maximilian Wich, and Georg Groh. 2021. Understanding and interpreting the impact of user context in hate speech detection. In *Proceedings of the Ninth International Workshop on Natural Language Processing for Social Media*, pages 91–102.

Maximilian Mozes, Pontus Stenetorp, Bennett Kleinberg, and Lewis Griffin. 2021. Frequency-guided word substitutions for detecting textual adversarial examples. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 171–186, Online. Association for Computational Linguistics.

Mesut Ozdag. 2018. Adversarial attacks and defenses against deep neural networks: A survey. *Procedia Computer Science*, 140:152–161. Cyber Physical Systems and Deep Learning Chicago, Illinois November 5-7, 2018.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. GloVe: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, Doha, Qatar. Association for Computational Linguistics.

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

Shuhuai Ren, Yihe Deng, Kun He, and Wanxiang Che. 2019. Generating natural language adversarial examples through probability weighted word saliency. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics.

Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. 2016. Why should i trust you?: Explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, pages 1135–1144.

Kevin Roth, Yannic Kilcher, and Thomas Hofmann. 2019. The odds are odd: A statistical test for detecting adversarial examples. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 5498–5507. PMLR.

David E Rumelhart, Geoffrey E Hinton, and Ronald J Williams. 1985. Learning internal representations by error propagation. Technical report, California Univ San Diego La Jolla Inst for Cognitive Science.

M. Schuster and K. K. Paliwal. 1997. Bidirectional recurrent neural networks. *IEEE Transactions on Signal Processing*, 45(11):2673–2681.

Lloyd S. Shapley. 1952. *A Value for n-Person Games*. RAND Corporation, Santa Monica, CA.

P. Sharma, D. Austin, and H. Liu. 2019. Attacks on machine learning: Adversarial examples in connected and autonomous vehicles. In *2019 IEEE International Symposium on Technologies for Homeland Security (HST)*, pages 1–7.

Avanti Shrikumar, Peyton Greenside, and Anshul Kundaje. 2017. Learning important features through propagating activation differences. In *Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6-11 August 2017*, volume 70 of *Proceedings of Machine Learning Research*, pages 3145–3153. PMLR.

Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. 2014. Deep inside convolutional networks: Visualising image classification models and saliency maps. In *2nd International Conference on Learning Representations, ICLR 2014*.

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

Nitish Srivastava, Geoffrey E. Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: a simple way to prevent neural networks from overfitting. *J. Mach. Learn. Res.*, 15(1):1929–1958.

Mukund Sundararajan, Ankur Taly, and Qiqi Yan. 2017. Axiomatic attribution for deep networks. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 3319–3328. JMLR. org.

Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. 2014. Intriguing properties of neural networks. In *International Conference on Learning Representations*.

Guanhong Tao, Shiqing Ma, Yingqi Liu, and Xiangyu Zhang. 2018. Attacks meet interpretability: Attribute-steered detection of adversarial samples. In *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc.

Xiaosen Wang, Hao Jin, and Kun He. 2019. Natural language adversarial attacks and defenses in word level. *arXiv preprint arXiv:1909.06723*.

Maximilian Wich, Edoardo Mosca, Adrian Gorniak, Johannes Hingerl, and Georg Groh. 2021. Explainable abusive language classification leveraging user and network data. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 481–496. Springer.

Weilin Xu, David Evans, and Yanjun Qi. 2018. Feature squeezing: Detecting adversarial examples in deep neural networks. In *Proceedings 2018 Network and Distributed System Security Symposium*. Internet Society.

Wei Emma Zhang, Quan Z. Sheng, Ahoud Alhazmi, and Chenliang Li. 2020. Adversarial attacks on deep-learning models in natural language processing. *ACM Transactions on Intelligent Systems and Technology*, 11(3):1–41.

Xiang Zhang, Junbo Zhao, and Yann LeCun. 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.

Yi Zhou, Xiaoqing Zheng, Cho-Jui Hsieh, Kai-wei Chang, and Xuanjing Huang. 2020. Defense against adversarial attacks in nlp via dirichlet neighborhood ensemble. *arXiv preprint arXiv:2006.11627*.

Yichao Zhou, Jyun-Yu Jiang, Kai-Wei Chang, and Wei Wang. 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.

B.4 STUDY VIII

Edoardo Mosca, Daryna Dementieva, Tohid Ebrahim Ajdari, Maximilian Kummeth, Kirill Gringauz, and Georg Groh (2023). "IFAN: An Explainability-Focused Interaction Framework for Humans and NLP Models." In: *arXiv preprint arXiv:2303.03124*. (Accepted at AACL, Nov 2023). URL: https://arxiv.org/abs/2303.03124

[3] https://creativecommons.org/licenses/by/4.0/

Publication Summary

"Interpretability and human oversight are fundamental pillars of deploying complex NLP models into real-world applications. However, applying explainability and human-in-the-loop methods requires technical proficiency. Despite existing toolkits for model understanding and analysis, options to integrate human feedback are still limited. We propose IFAN, a framework for real-time explanation-based interaction with NLP models. Through IFAN's interface, users can provide feedback to selected model explanations, which is then integrated through adapter layers to align the model with human rationale. We show the system to be effective in debiasing a hate speech classifier with minimal performance loss. IFAN also offers a visual admin system and API to manage models (and datasets) as well as control access rights. A demo is live at ifan.ml." (Mosca, Dementieva, et al., 2023, p.1)

Author Contributions

Edoardo Mosca contributed to the study as follows:

- Conception, development, and lead of the research project **80%**
- Development of the platform **70%**
- Experimentation on use cases **40%**
- Drafting of the manuscript **50%**
- Submission, peer review, and publication process **60%**

IFAN: An Explainability-Focused Interaction Framework for Humans and NLP Models

Edoardo Mosca, Daryna Dementieva, Tohid Ebrahim Ajdari,
Maximilian Kummeth, Kirill Gringauz and **Georg Groh**
TU Munich, Department of Informatics, Germany
`{name.surname}@tum.de`
`grohg@in.tum.de`

Abstract

Interpretability and human oversight are fundamental pillars of deploying complex NLP models into real-world applications. However, applying explainability and human-in-the-loop methods requires technical proficiency. Despite existing toolkits for model understanding and analysis, options to integrate human feedback are still limited. We propose IFAN, a framework for real-time explanation-based interaction with NLP models. Through IFAN's interface, users can provide feedback to selected model explanations, which is then integrated through adapter layers to align the model with human rationale. We show the system to be effective in debiasing a hate speech classifier with minimal performance loss. IFAN also offers a visual admin system and API to manage models (and datasets) as well as control access rights. A demo is live at ifan.ml.

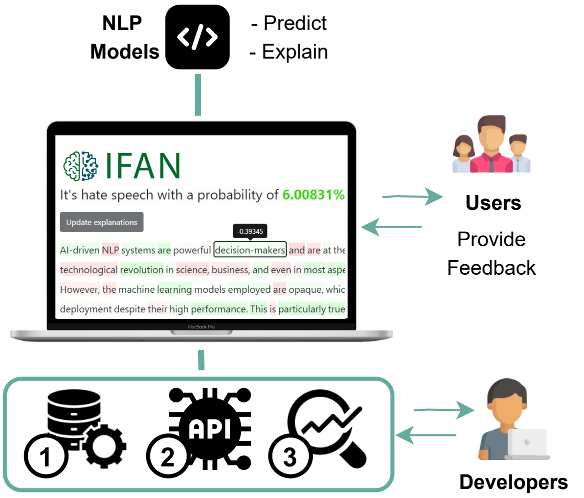

NLP Models
- Predict
- Explain
IFAN
It's hate speech with a probability of 6.00831%
Update explanations
-0.39345
AI-driven NLP systems are powerful decision-makers and are at the
technological revolution in science, business, and even in most aspe
However, the machine learning models employed are opaque, whic
deployment despite their high performance. This is particularly true
Users
Provide Feedback
API
1
2
3
Developers

Figure 1: IFAN in brief. The interface allows NLP models and users to interact through predictions, explanations, and feedback. IFAN also provides developers with (1) a manager for models and datasets, (2) model API access, and (3) reports about the model.

1 Introduction

As *Natural Language Processing* (NLP) systems continue to improve in performance, they are increasingly adopted in real-world applications (Khurana et al., 2022). *Large Language Models* (LLMs)—such as GPT-3 (Brown et al., 2020), BLOOM (Scao et al., 2022), and T5 (Raffel et al., 2020)—are without a shred of doubt the main protagonists of recent advances in the field. They are able to substantially outperform previous solutions while being directly applicable to any NLP task.

There are however strong concerns given the black-box nature of such architectures (Madsen et al., 2022; Mosca et al., 2022a). In fact, their large scale and high complexity are substantial drawbacks in terms of *transparency*, *accountability*, and *human oversight*. Beyond ethical considerations, even legal guidelines from the European Union are now explicitly defining these interpretability factors as essential for any deployed AI system (European Commission, 2020).

Research efforts in *eXplainable Artificial Intelligence* (XAI) (Arrieta et al., 2020; Mosca et al., 2022b) and *Human-in-the-Loop* (HitL) machine learning (Monarch, 2021) have thus been on the rise—producing solutions that aim at mitigating the current lack of interpretability. Most notably, the recent literature contains a number of toolkits and frameworks to analyze, understand, and improve complex NLP models (Wallace et al., 2019; Liu et al., 2021). Some of them even offer low-code interfaces for stakeholders who do not possess the otherwise required technical proficiency. Nonetheless, current options to collect human rationale and provide it as feedback to the model are still limited.

We propose IFAN, a novel low-to-no-code framework to interact in real time with NLP models via explanations. Our contribution can be summarized as follows:

(1) IFAN offers an interface for users to provide feedback to selected model explanations, which is then integrated via parameter-efficient adapter layers.

(2) Our live platform also offers a visual administration system and API to manage models, datasets, and users as well as their corresponding access rights.

(3) We show the efficiency of our framework in debiasing a hate speech classifier and propose a feedback-rebalancing step to mitigate the model's forgetfulness across updates.

IFAN's demo is accessible at ifan[1].ml together with its documentation.[2] Full access is available with login credentials, which we can provide upon request. A supplementary video showcase can be found online[3].

2 Related Work

2.1 HitL with Model Explanations

Human-in-the-Loop (HitL) machine learning studies how models can be continuously improved with human feedback (Monarch, 2021). While a large part of the HitL literature deals with label-focused feedback such as *active learning*, more recent works explore how explanations can be leveraged to provide more detailed human rationale (Lertvittayakumjorn and Toni, 2021).

Combining classical HitL (Wang et al., 2021) with explanations to construct human feedback for the model (Han et al., 2020) has been referred to as *Explanation-Based Human Debugging* (EBHD) (Lertvittayakumjorn and Toni, 2021). Good examples are Ray et al. (2019), Selvaraju et al. (2019), and Strout et al. (2019), which show improvements in performance and interpretability when iteratively providing models with human rationale.

A more NLP-focused EBHD approach is Yao et al. (2021), where the authors leverage explanations to debug and refine two transformer instances—BERT (Devlin et al., 2019) and RoBERTa (Liu et al., 2019). Concretely, word saliency explanations at different levels of granularity are provided to humans, who in turn provide suggestions in the form of natural language. The annotator's feedback is converted into first-order logic rules, which are later utilized to condition learning with new samples.

2.2 Interactive NLP Analysis Platforms

In the recent literature, we can find strong contributions in terms of software and digital toolkits to analyze and explain NLP models (Wallace et al., 2019; Hoover et al., 2020) as well as further refining them via parameter-efficient fine-tuning (Beck et al., 2022).

For instance, Liu et al. (2021) proposes EXPLAINABOARD, an interactive explainability-focused leaderboard for NLP models. More in detail, it allows researchers to run diagnostics about the strengths and weaknesses of a given model, compare different architectures, and closely analyze predictions as well as recurring model mistakes. Similarly, the LANGUAGE INTERPRETABILITY TOOL by Tenney et al. (2020) is an open-source platform and API to visualize and understand NLP models. In particular, it provides a browser-based interface integrating local explanations as well as counterfactual examples to enable model interpretability and error analysis.

Finally, Beck et al. (2022) releases ADAPTERHUB PLAYGROUND, a no-code platform to few-shot learning with language models. Specifically, the authors built an intuitive interface where users can easily perform predictions and training of complex NLP models on several natural language tasks.

3 IFAN

The **I**nteraction **F**ramework for **A**rtificial and **N**atural Intelligence (**IFAN**) is a web-based platform for inspecting and controlling text processing models. Its main goal is to decrease the opacity of NLP systems and integrate explanation-based HitL into their development pipeline. Through our interface, stakeholders can test and explain models' behavior and—when encountering anomalies in predictions or explanations—they can fix them onsite by providing feedback.

The main blocks of the platform are presented in Figure 2. The **Backbone** part contains all machine learning development components—datasets and models. We adopt HuggingFace formats (see 3.3 and 3.4) (Wolf et al., 2020) and wrap the entire backbone as a Docker[4] image for deployment. The **User Interface** is the visual component of the platform, where all the human-machine interaction takes place. Here, developers have also access to additional visual resources to configure details about models, datasets, and users.

[1] https://ifan.ml

[2] https://ifan.ml/documentation

[3] https://www.youtube.com/watch?v=BzzoQzTsrLo

[4] https://www.docker.com

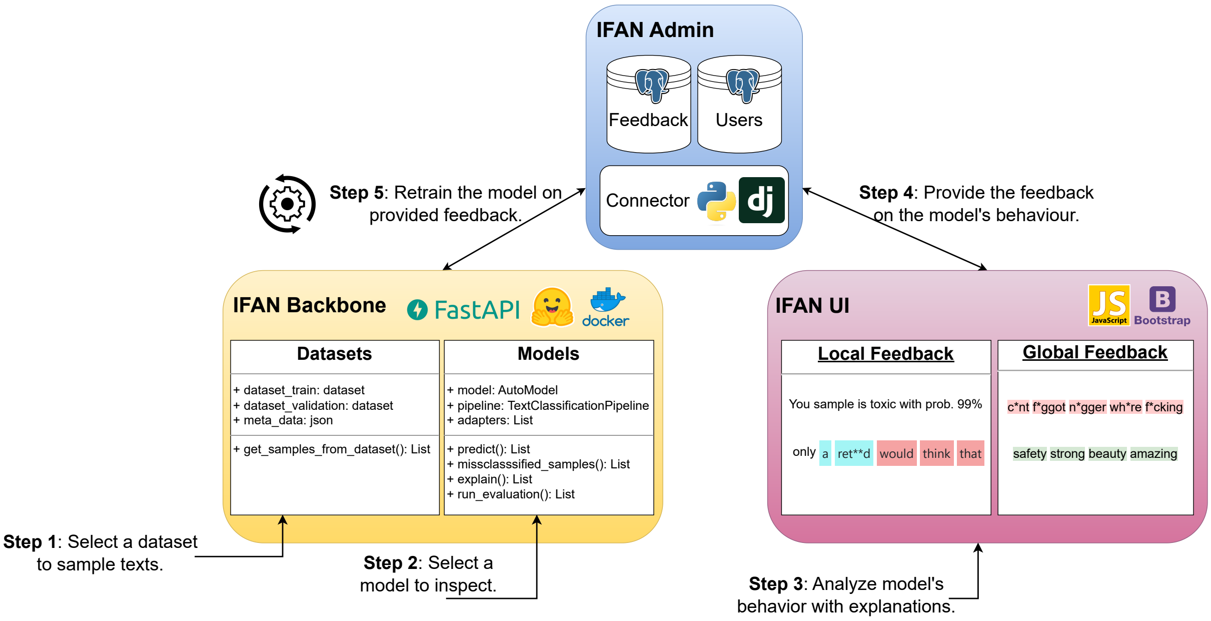

IFAN Admin
Feedback
Users
Connector
Step 5: Retrain the model on provided feedback.
Step 4: Provide the feedback on the model's behaviour.
IFAN Backbone
FastAPI
docker
Datasets
+ dataset_train: dataset
+ dataset_validation: dataset
+ meta_data: json
+ get_samples_from_dataset(): List
Models
+ model: AutoModel
+ pipeline: TextClassificationPipeline
+ adapters: List
+ predict(): List
+ missclassified_samples(): List
+ explain(): List
+ run_evaluation(): List
IFAN UI
JS JavaScript
B Bootstrap
Local Feedback
You sample is toxic with prob. 99%
only a ret**d would think that
Global Feedback
c*nt f*ggot n*gger wh*re f*cking
safety strong beauty amazing
Step 1: Select a dataset to sample texts.
Step 2: Select a model to inspect.
Step 3: Analyze model's behavior with explanations.

Figure 2: Overall schema of IFAN idea: (i) The user selects a dataset or writes a customized input. (ii) Then the user can select a model which should be inspected. (iii) With the UI, annotators can check the model's prediction on a sample and two types of explanations – local and global. (iv) If there is some misbehavior, the annotators can provide feedback. (iv) The feedback is stored and then used to fine-tune the model.

The connection between the backbone and the user interface is managed by the **Admin** component. All the user data and rights as well as samples receiving feedback are stored in a PostgreSQL[5] database instance. The communication is handled via Python Django[6], which integrates everything w.r.t. user authentication, API calls/responses, state logs, and location of backbone resources. In the next sections, we provide a more detailed description of the main platform components.

3.1 User Interface

Our frontend is built with Boostrap[7] and JavaScript[8]. Currently, the pages available in our UI are the following:

Landing Page Here users can get a short introduction to IFAN. We briefly explain our platform's goals, the concept of HitL, and how our framework can be integrated into the development of NLP models.

Documentation It provides a detailed description of all the UI components together with screenshots and guidelines. Here, users can find specific instructions on how to configure and interact with our platform.

Feedback This is the main interaction page. Here, users can run a model on an input sample either taken from the dataset or that they wrote themselves. Then, they can load the model's prediction and explanations and provide feedback both in terms of re-labeling and adjusting each feature's relevance.

Configuration This page has limited access (see 3.2). Here, developers can configure and manage the platform, More specifically, users can be created, modified, and deleted as well as upgraded or downgraded in their roles and access rights. Also, they can manage models and datasets as well as specify the currently active ones.

Account Settings Each authorized user can view, edit, export, and delete their account data as well as reset their login password.

3.2 Users

The platform separates users in three tiers: *developers*, *annotators*, and *unauthorized* users (Table 1).

Unauthorized users do not possess login credentials and have limited access to the platform. They can visualize model predictions and explanations but their feedback is not considered.

[5] https://www.postgresql.org
[6] https://www.djangoproject.com
[7] https://getbootstrap.com
[8] https://www.javascript.com

	Dev	Annotator	Unauthorized
Classification & Explanations	✓	✓	✓
Smart Samples Selection	✓	✓	✗
Feedback	✓	✓	✗
Active Configuration	✓	✗	✗
New Models & Datasets Upload	✓	✗	✗
New Users Creation	✓	✗	✗

Table 1: Different levels of access to IFAN functionalities.

Normal users (or annotators) are known through their credentials and can thus actively engage with the model. During a HitL iteration, they can use the feedback page with pre-configured datasets and models, test the model on a text sample, view explanations, and provide feedback if needed.

Developers have full access and can configure all aspects of the platform. More specifically, they have access to the *configuration* page (see 3.1) and can thus manage anything regarding users, roles, API access, models, and datasets.

3.3 Datasets

Before the model's behavior exploration, the *active dataset* should be specified via the configuration page (see 3.1). This is the dataset from which the text examples for the model testing are sampled.

Dataset	**Short Description**
HateXplain (Mathew et al., 2021)	A dataset for hate speech classification which has 3 classes for hate type detection, the target community classification, and rationales.
GYAFC (Rao and Tetreault, 2018)	Formality detection dataset which corresponds to 2-class classification: formal and informal.

Table 2: Example of datasets available at IFAN for testing.

We conform to a standard format by using the HuggingFace Datasets library[9]. Developers interacting with our platform are strongly encouraged to adhere to this standard when uploading new datasets and making them available to the interface. Table 2 shows two examples of datasets already available on our platform.

3.4 Models

Analogous to datasets, our platform specifies an *active model* at any time and the employed models adhere to the standard used by HuggingFace Models[10].

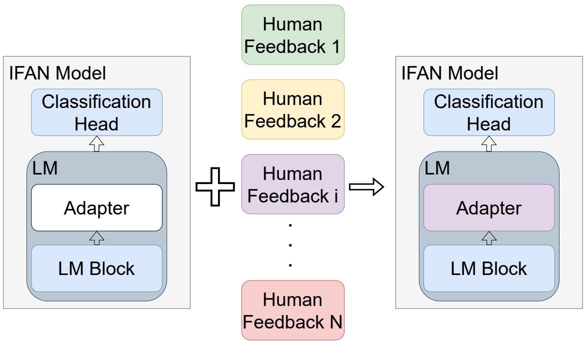


Figure 3: The proposed architecture for the models integrated into IFAN: addition of Adapter layer which is trainable on provided human feedback.

To incorporate feedback into our models, we utilize adapter layers (Houlsby et al., 2019), a parameter-efficient fine-tuning technique. Figure 3 sketches an overview of the architecture used. Adapters are integrated on top of each language model unit (e.g. transformer block) and are trained with the human feedback while we freeze the rest of the model's weights. Adapters can also be disabled at any time to recover to the original state of the model.

3.5 Explanations & Feedback Mechanism

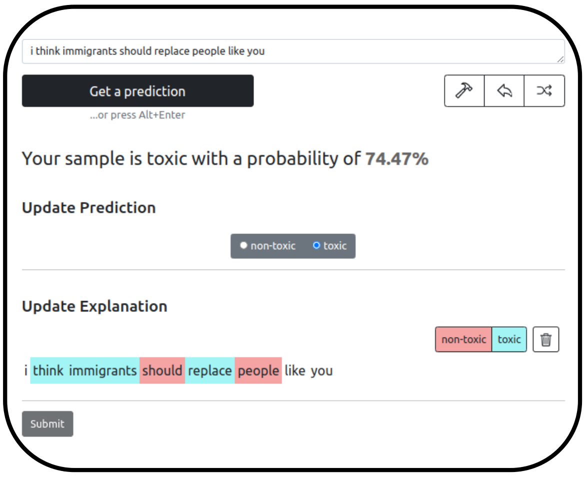
Users can evaluate the active model on the active dataset through the Feedback page. They may input text in three ways: i) create a text sample themselves; if authorized: ii) sample a random text from the active dataset; iii) sample a random *misclassified* text from the test part of the active dataset. Users receive the classification results and the model's confidence. They can assess the result and correct any misclassifications.

To further inspect the model's behavior, we provide two types of explanations—local and global. For local explanations on a text sample, we display relevant features to each output class (Figure 4). We attribute scores using the LIME framework (Ribeiro et al., 2016) and—to filter weak correlations—we highlight as relevant only tokens with a score above the threshold $\theta = 0.1$. On the global side, we list the most influential unigrams

[9] https://huggingface.co/docs/datasets/index

[10] https://huggingface.co/docs/transformers/main_classes/model

for each output class. These can be inspected to extract insights about what keywords and patterns the model focuses on at the dataset level. For all 1-grams present in a dataset, their corresponding classification scores are calculated and the tokens with top scores are displayed on the page.


i think immigrants should replace people like you

Your sample is toxic with a probability of 74.47%

Update Prediction

● non-toxic ○ toxic

Update Explanation

i think immigrants should replace people like you

Figure 4: The example of the results and local explanations that annotators can obtain on the Feedback page.

Annotators can easily edit the highlighted tokens and send the updated explanation as feedback. We store the result—i.e. the highlighted relevant parts—and use them to fine-tune the adapter layers (see 3.4).

Regarding the fine-tuning procedure, directly using the highlighted feedback text for adapter fine-tuning causes significant losses in the original model performance. We propose to mix feedback with original samples to mitigate this effect, which allows effective feedback incorporation while reducing model forgetfulness. See 4 for more details.

3.6 Backbone API

We expose our backbone's API to make available all essential dataset/model management functions. These provide a high-level interface for additional experiments dealing with model evaluation, explanation, and feedback. The API was built with the Python framework FastAPI[11], detailed screenshots can be found in the Appendix A.

4 Case Study

We carried out a case study to test the applicability of IFAN. We chose a hate speech detection task based on the HateXplain dataset (Mathew et al., 2021). The goal of the experiment was to use our framework to debias a given hate speech detector.

Firstly, we modified the original dataset for binary classification task—*"toxic"* and *"non-toxic"*—and fine-tuned a BERT model (Devlin et al., 2019) (BERT-Tiny uncased snapshot)[12]. We choose the Jewish subgroup as a target for our debiasing process.

We annotate 24 random misclassified samples, 12 with the most confidence and 12 with the least confidence scores (see Appendix C.1). We invited 3 annotators to participate in the annotation process. The n-grams that were modified by annotators were saved and used to create a new training dataset for the adapters. As a result, we collected 40 annotated n-grams and repeated them to get 120 training samples. To complete the new training creation, we balanced these samples with 500 original samples (250 toxic, 250 non-toxic) randomly selected from the HateXplain dataset.

Model	**Pr**	**Re**	**F1**	**Pr$_J$**
BERT (baseline)	0.80	**0.78**	**0.79**	0.95
Most Confident Missclassified				
BERT+Feedback (non-bal.)	0.34	0.28	0.31	0.82
BERT+Feedback (bal.)	0.78	0.80	**0.79**	**0.97**
Least Confident Missclassified				
BERT+Feedback (non-bal.)	**0.83**	0.73	0.78	0.96
BERT+Feedback (bal.)	0.79	**0.78**	0.78	0.96

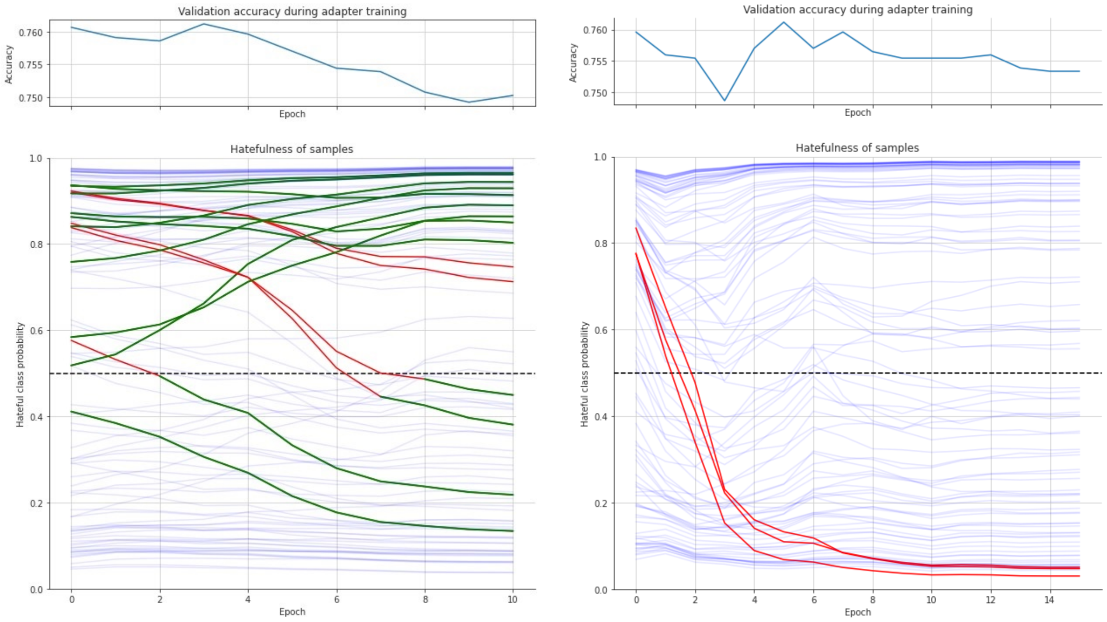
Table 3: The results of the case study: hate speech classification model debiasing. We compare different strategies for feedback incorporation. Pr$_J$ states for the Precision score on the Jewish target group.

The results are presented in Table 3. We observe that the non-balanced training dataset, which only contains feedback on the most confidently misclassified samples, resulted in a significant decrease in performance. While the inclusion of feedback on least confident samples caused a slight decline in the overall F1 score, Adapter training on the balanced feedback led to an improvement in the precision score for the Jewish target group.

Figure 5 shows the changes in the detector while fine-tuning with the collected feedback. When rebalancing the feedback, only modified samples are drastically changed while the performance on the original texts is only slightly affected. A more detailed comparison between fine-tuning on non-balanced and balanced feedback can be found in Appendix C.2.

[11] https://fastapi.tiangolo.com

[12] https://huggingface.co/google/bert_uncased_L-2_H-128_A-2



(a) Training on feedback on the Jewish subgroup samples.

(b) Training on feedback samples with "jewish" key-words.

Figure 5: The results of the domain case using IFAN platform. We can observe that for both experiments with balanced training data, the overall model's performance is only slightly changed while the model's behavior on the Jewish target group is improved.

5 Limitations & Future Work

As of now, our feedback system is limited to applications in the sequence-to-class format. However, current and future work is already focusing on extending the pipeline to token-to-class and sequence-to-sequence use cases.

At the same time, we currently offer a limited set of explanation, feedback, and management options, which we plan to increase in the immediate future. A small user study has been conducted (Appendix B) to collect feedback about the platform and improve its user-friendliness. Our intent is to continue iterating the development of new features with trials with developers and laymen.

Finally, our experiments do not yet show clear trends w.r.t. the correlation between performance and feedback hyperparameters. Indeed, further research and trials have to be carried out to establish optimal choices for the number of feedback samples, fine-tuning epochs, and the rebalancing ratio.

6 Conclusion

This work proposes IFAN, a framework focusing on real-time explanation-based interaction between NLP models and human annotators. Our contribution is motivated by the limited options in terms of existing tools to interpret and control NLP models.

IFAN is composed of three main units. The **Backbone** unifies all the machine learning pipelines and exposes an API for accessibility. The **User Interface**—organized in *landing page*, *documentation*, *feedback*, and *configuration*—provides an intuitive visual component to interact with models. Finally, the **Admin** controls the connection between the two previous components.

Additionally, we introduce the feedback mechanism that takes advantage of adapter layers to efficiently and iteratively fine-tune models on the downstream task. Our experiments show the frameworks' credibility at debiasing a hate speech classifier with minimal performance loss.

We believe IFAN to be a valuable step towards enabling the interpretable and controllable deployment of NLP models—allowing users with no technical proficiency to interact and provide feedback to deployed NLP systems. Regarding future work, we set as a priority to extend the framework to more NLP tasks as well as to integrate additional model analysis features and feedback mechanisms.

Acknowledgments

We thank Yutong Zhou for her valuable contribution to our user interface. This paper has been supported by the *German Federal Ministry of Education and Research* (BMBF, grant 01IS17049).

Ethical Considerations

Interpretability and controllability of modern NLP models and systems are fundamental pillars for their ethical and safe deployment (European Commission, 2020). This works aims at having a positive impact on both aspects as it provides a tool to explain models and provide them with feedback. By reducing the technical proficiency required to interact with NLP systems, we hope to facilitate the process of providing valuable human rationales to influence complex models. We strongly encourage future work to keep exploring this research direction as it enables to involve a larger and more diverse crowd, thus positively affecting also other desiderata such as *fairness*, *transparency*, and *accountability*. Nevertheless, there are potential pitfalls worth considering.

Ensuring high quality for the human feedback is challenging (Al Kuwatly et al., 2020), and exposing models to external influence can be used as an exploit by adversarial agents (Mosca et al., 2022a). Especially with a very small crowd of annotators, there's potential for a few people to have a strong influence on the model. A restrictive access rights management system like IFAN's already mitigates these issues. We believe that additional security features as well as tracking annotators' impact are key for future work to foster their trustworthiness.

Previous works mention that users can feel discouraged and frustrated when interacting with poor models and badly-designed interfaces, which can also affect feedback quality (Lertvittayakumjorn and Toni, 2021). This can be addressed by integrating user studies in the development process in order to design more intuitive interfaces and improve the overall user experience.

On the opposite end of the spectrum, plausible explanations can make humans overestimate the model's capabilities and make them trust systems that are still not ready for deployment. In this case, a more diverse and complementary set of explanations for users (Madsen et al., 2022) as well as comprehensive model reports for developers are core goals to provide a more complete picture of the models to be deployed.

References

Hala Al Kuwatly, Maximilian Wich, and Georg Groh. 2020. Identifying and measuring annotator bias based on annotators' demographic characteristics. In *Proceedings of the Fourth Workshop on Online Abuse and Harms*, pages 184–190, Online. Association for Computational Linguistics.

Alejandro Barredo Arrieta, Natalia Díaz-Rodríguez, Javier Del Ser, Adrien Bennetot, Siham Tabik, Alberto Barbado, Salvador García, Sergio Gil-López, Daniel Molina, Richard Benjamins, et al. 2020. Explainable artificial intelligence (xai): Concepts, taxonomies, opportunities and challenges toward responsible ai. *Information Fusion*, 58:82–115.

Tilman Beck, Bela Bohlender, Christina Viehmann, Vincent Hane, Yanik Adamson, Jaber Khuri, Jonas Brossmann, Jonas Pfeiffer, and Iryna Gurevych. 2022. AdapterHub playground: Simple and flexible few-shot learning with adapters. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 61–75, Dublin, Ireland. Association for Computational Linguistics.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

European Commission. 2020. White paper on artificial intelligence: a european approach to excellence and trust. *Com (2020) 65 Final*.

Xiaochuang Han, Byron C. Wallace, and Yulia Tsvetkov. 2020. Explaining black box predictions and unveiling data artifacts through influence functions. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5553–5563, Online. Association for Computational Linguistics.

Benjamin Hoover, Hendrik Strobelt, and Sebastian Gehrmann. 2020. exBERT: A Visual Analysis Tool to Explore Learned Representations in Transformer Models. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 187–196, Online. Association for Computational Linguistics.

Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin de Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. 2019. Parameter-efficient transfer learning for NLP. In *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA*, volume 97 of

Proceedings of Machine Learning Research, pages 2790–2799. PMLR.

Diksha Khurana, Aditya Koli, Kiran Khatter, and Sukhdev Singh. 2022. Natural language processing: State of the art, current trends and challenges. *Multimedia tools and applications*, pages 1–32.

Piyawat Lertvittayakumjorn and Francesca Toni. 2021. Explanation-based human debugging of NLP models: A survey. *Transactions of the Association for Computational Linguistics*, 9:1508–1528.

Pengfei Liu, Jinlan Fu, Yang Xiao, Weizhe Yuan, Shuaichen Chang, Junqi Dai, Yixin Liu, Zihuiwen Ye, and Graham Neubig. 2021. ExplainaBoard: An explainable leaderboard for NLP. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing: System Demonstrations*, pages 280–289, Online. Association for Computational Linguistics.

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

Andreas Madsen, Siva Reddy, and Sarath Chandar. 2022. Post-hoc interpretability for neural nlp: A survey. *ACM Comput. Surv.*, 55(8).

Binny Mathew, Punyajoy Saha, Seid Muhie Yimam, Chris Biemann, Pawan Goyal, and Animesh Mukherjee. 2021. Hatexplain: A benchmark dataset for explainable hate speech detection. In *Thirty-Fifth AAAI Conference on Artificial Intelligence, AAAI 2021, Thirty-Third Conference on Innovative Applications of Artificial Intelligence, IAAI 2021, The Eleventh Symposium on Educational Advances in Artificial Intelligence, EAAI 2021, Virtual Event, February 2-9, 2021*, pages 14867–14875. AAAI Press.

Robert Munro Monarch. 2021. *Human-in-the-Loop Machine Learning: Active learning and annotation for human-centered AI*. Simon and Schuster.

Edoardo Mosca, Shreyash Agarwal, Javier Rando Ramírez, and Georg Groh. 2022a. “that is a suspicious reaction!”: Interpreting logits variation to detect NLP adversarial attacks. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 7806–7816, Dublin, Ireland. Association for Computational Linguistics.

Edoardo Mosca, Ferenc Szigeti, Stella Tragianni, Daniel Gallagher, and Georg Groh. 2022b. SHAP-based explanation methods: A review for NLP interpretability. In *Proceedings of the 29th International Conference on Computational Linguistics*, pages 4593–4603, Gyeongju, Republic of Korea. International Committee on Computational Linguistics.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *J. Mach. Learn. Res.*, 21:140:1–140:67.

Sudha Rao and Joel Tetreault. 2018. Dear sir or madam, may I introduce the GYAFC dataset: Corpus, benchmarks and metrics for formality style transfer. 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 129–140, New Orleans, Louisiana. Association for Computational Linguistics.

Arijit Ray, Yi Yao, Rakesh Kumar, Ajay Divakaran, and Giedrius Burachas. 2019. Can you explain that? lucid explanations help human-ai collaborative image retrieval. In *Proceedings of the AAAI Conference on Human Computation and Crowdsourcing*, volume 7, pages 153–161.

Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. 2016. " why should i trust you?" explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, pages 1135–1144.

Teven Le Scao, Angela Fan, Christopher Akiki, Ellie Pavlick, Suzana Ilic, Daniel Hesslow, Roman Castagné, Alexandra Sasha Luccioni, François Yvon, Matthias Gallé, Jonathan Tow, Alexander M. Rush, Stella Biderman, Albert Webson, Pawan Sasanka Ammanamanchi, Thomas Wang, Benoît Sagot, Niklas Muennighoff, Albert Villanova del Moral, Olatunji Ruwase, Rachel Bawden, Stas Bekman, Angelina McMillan-Major, Iz Beltagy, Huu Nguyen, Lucile Saulnier, Samson Tan, Pedro Ortiz Suarez, Victor Sanh, Hugo Laurençon, Yacine Jernite, Julien Launay, Margaret Mitchell, Colin Raffel, Aaron Gokaslan, Adi Simhi, Aitor Soroa, Alham Fikri Aji, Amit Alfassy, Anna Rogers, Ariel Kreisberg Nitzav, Canwen Xu, Chenghao Mou, Chris Emezue, Christopher Klamm, Colin Leong, Daniel van Strien, David Ifeoluwa Adelani, and et al. 2022. BLOOM: A 176b-parameter open-access multilingual language model. *CoRR*, abs/2211.05100.

Ramprasaath R Selvaraju, Stefan Lee, Yilin Shen, Hongxia Jin, Shalini Ghosh, Larry Heck, Dhruv Batra, and Devi Parikh. 2019. Taking a hint: Leveraging explanations to make vision and language models more grounded. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2591–2600.

Julia Strout, Ye Zhang, and Raymond Mooney. 2019. Do human rationales improve machine explanations? In *Proceedings of the 2019 ACL Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 56–62, Florence, Italy. Association for Computational Linguistics.

Ian Tenney, James Wexler, Jasmijn Bastings, Tolga Bolukbasi, Andy Coenen, Sebastian Gehrmann, Ellen Jiang, Mahima Pushkarna, Carey Radebaugh, Emily Reif, and Ann Yuan. 2020. The language interpretability tool: Extensible, interactive visualizations and analysis for NLP models. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 107–118, Online. Association for Computational Linguistics.

Eric Wallace, Jens Tuyls, Junlin Wang, Sanjay Subramanian, Matt Gardner, and Sameer Singh. 2019. AllenNLP interpret: A framework for explaining predictions of NLP models. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP): System Demonstrations*, pages 7–12, Hong Kong, China. Association for Computational Linguistics.

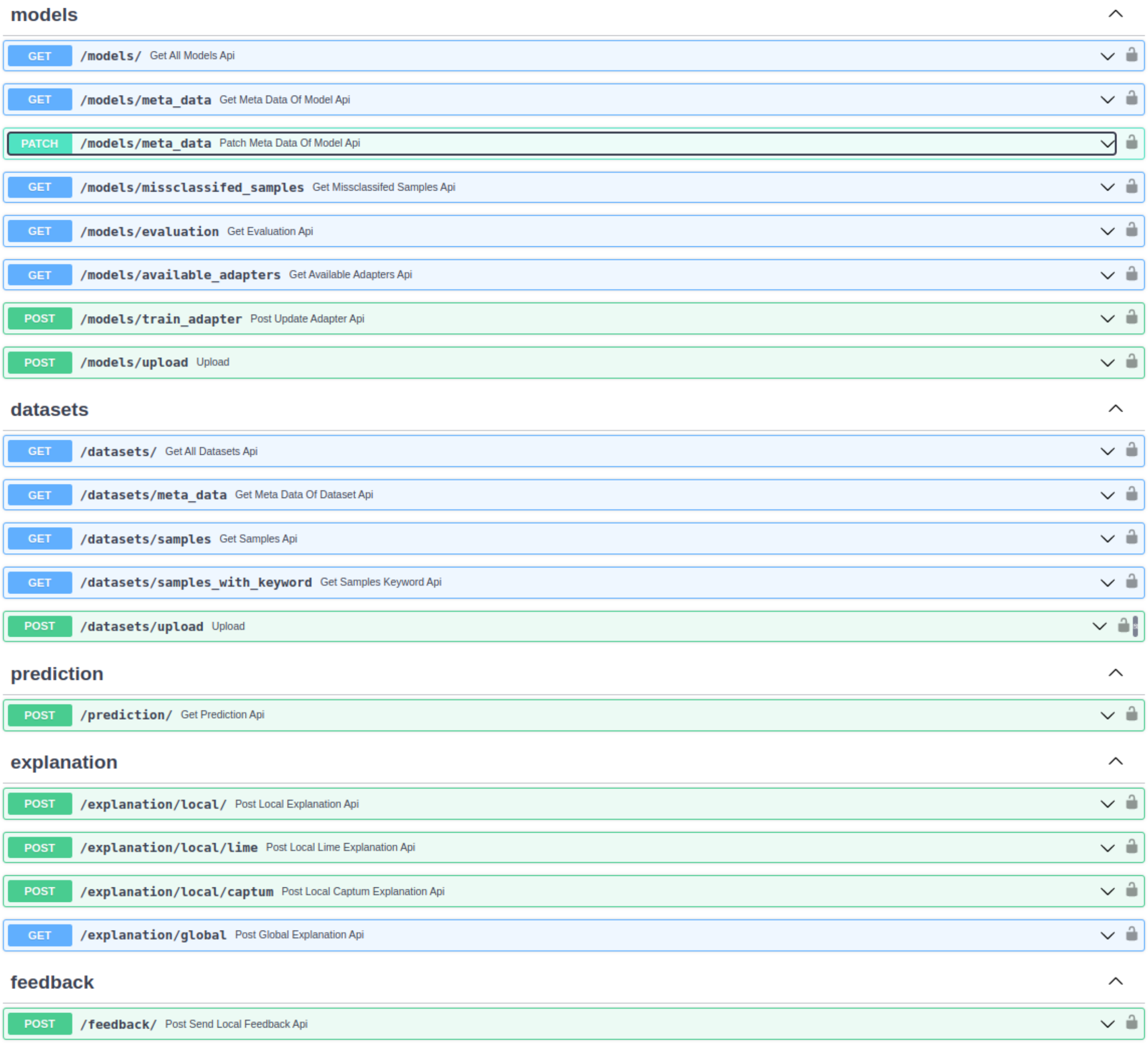
Zijie J. Wang, Dongjin Choi, Shenyu Xu, and Diyi Yang. 2021. Putting humans in the natural language processing loop: A survey. In *Proceedings of the First Workshop on Bridging Human–Computer Interaction and Natural Language Processing*, pages 47–52, Online. Association for Computational Linguistics.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

Huihan Yao, Ying Chen, Qinyuan Ye, Xisen Jin, and Xiang Ren. 2021. Refining language models with compositional explanations. *Advances in Neural Information Processing Systems*, 34:8954–8967.

A Backbone API Endpoints

Figure 6 shows the auto-generated docs for our backbone's REST API, which serves as guidelines to interact with our backbone. Endpoints are divided into functional groups—*models*, *datasets*, *prediction*, *explanation*, and *feedback*). Currently, this page is only accessible within our institution's network for security reasons.



models

GET /models/ Get All Models Api

GET /models/meta_data Get Meta Data Of Model Api

PATCH /models/meta_data Patch Meta Data Of Model Api

GET /models/missclassifed_samples Get Missclassifed Samples Api

GET /models/evaluation Get Evaluation Api

GET /models/available_adapters Get Available Adapters Api

POST /models/train_adapter Post Update Adapter Api

POST /models/upload Upload

datasets

GET /datasets/ Get All Datasets Api

GET /datasets/meta_data Get Meta Data Of Dataset Api

GET /datasets/samples Get Samples Api

GET /datasets/samples_with_keyword Get Samples Keyword Api

POST /datasets/upload Upload

prediction

POST /prediction/ Get Prediction Api

explanation

POST /explanation/local/ Post Local Explanation Api

POST /explanation/local/lime Post Local Lime Explanation Api

POST /explanation/local/captum Post Local Captum Explanation Api

GET /explanation/global Post Global Explanation Api

feedback

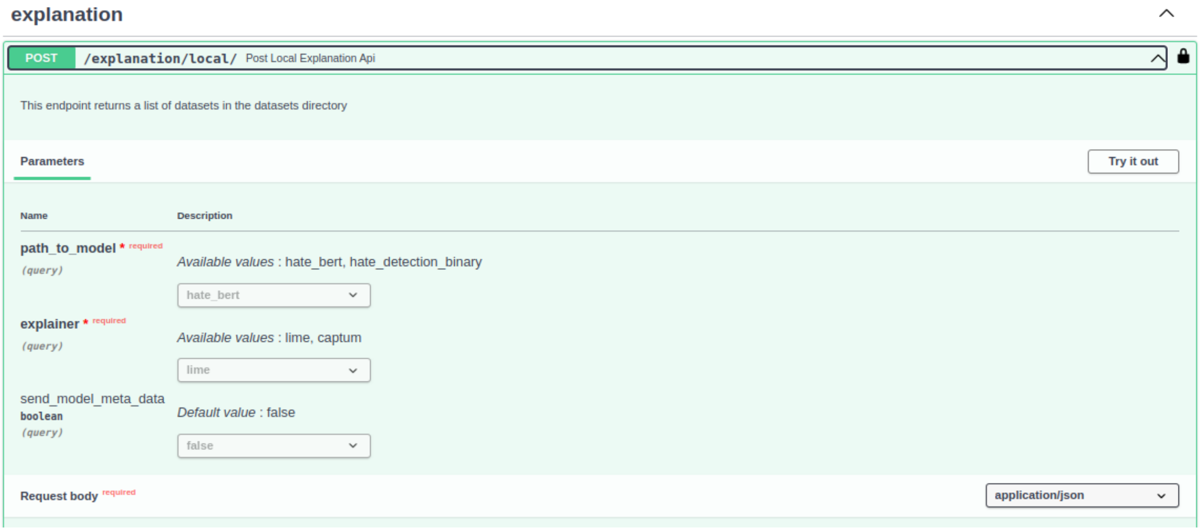
POST /feedback/ Post Send Local Feedback Api

Figure 6: Screenshot of the Swagger UI for our backbone API endpoints.

Developers with direct API access (specifiable on the *configuration page*, see 3.1) can directly make requests to this high-level interface for additional (larger-scale) experiments. Once again, the API has been built with the Python framework FastAPI[13].

Figure 7 shows the documentation for the *explanation* endpoint. Here, we can inspect the details about the endpoint, such as the required parameters—i.e. the path to the model, the explainer to be used (e.g. LIME), and the model's prediction as body request.

[13] https://fastapi.tiangolo.com



explanation

POST /explanation/local/ Post Local Explanation Api

This endpoint returns a list of datasets in the datasets directory

Parameters

Name	Description
path_to_model * required (query)	*Available values* : hate_bert, hate_detection_binary hate_bert
explainer * required (query)	*Available values* : lime, captum lime
send_model_meta_data boolean (query)	*Default value* : false false

Request body required

application/json

Figure 7: Screenshot of the *explanation* endpoint from our backbone API's Swagger UI.

B User Study

We performed a small user study evaluating the usability of our platform with users having different backgrounds. We gathered a total of nine testers, which we organize into three different categories based on their familiarity with the underlying methodologies. Users may be *laymen*, *computer scientists*, or *experts*. For each of these categories, three testers are assigned. Laymen are general users with no exposure to computer science. Computer scientists are those who studied computer science or a related discipline and who work in corresponding fields, such as software development. People who also possess knowledge in the domains of machine learning and NLP are considered experts.

We asked the users to try out to interact with IFAN and then answer several questions about the website's usability. The questionnaire was of the following structure:

- The first question is about the competency of the test person, which is categorized as computer science, natural language processing expert, or layman.

- Users assign a score between 1 and 5 to the local explanation on the left side of the feedback page.

- A textual input field for the users to hand in more detailed feedback on the previous rating.

- Users assign a score between 1 and 5 to the global explanation on the right side of the feedback page.

- A textual input field for the users to hand in more detailed feedback on the previous rating.

- Users rate the overall usability of the website with a score between 1 and 5.

- A textual input field for the users to hand in more detailed feedback on the previous rating.

The average ratings given by users: (i) Local explanation: 3.88; (ii) Global explanation: 3.2; (iii) General usability: 3.33. The problem that the users mentioned were insufficiency of only one form of explanations and feedback, sometimes misleading interface for markup, and waiting time. While some of the problems are already fixed in the current version of the platform, we are planning to consistently improve the usability of the platform. At the same time, we note a positive result—the users with different backgrounds were overall comfortable with IFAN usage and found it useful to increase the controllability of NLP models. We believe this confirms the possibility of IFAN utilization for stakeholders, developers, and annotators in future applications.

C Case Study

In this section, we provide additional information about the experiment conducted under Case Study (Section 4).

C.1 Annotated Samples

Here, we provide the list of that were used for manual annotation (Section 4). These samples were misclassified by hate speech detection model. In Table 4, we provide the misclassified samples with the lowest confidence, in Table 5 – with the highest confidence.

Sample	**Original Label**	**Prediction**	**Confidence Score**
speaking of does anyone know which goy group the jews intermarry with in the greatest numbers i was under the impression it was either asians or catholics	toxic	non-toxic	0.52
it not so much about the torah which is pretty innocuous and ignored by many jews despite what is said it really about that forbidden book the talmud or babylonian talmud which as goyim cattle we are not permitted to even read good luck with your research	non-toxic	toxic	0.56
if memory serves ashkenazi jews and east asians have higher average iqs	non-toxic	toxic	0.57
<user> the jews faked the moon landing	toxic	non-toxic	0.52
<user> <user> she will blame it on jews or trump i bet	toxic	non-toxic	0.53
christianity role is historical protector of the jews it has afforded the tribe the time and space it needed to prepare white genocide teamwhite	toxic	non-toxic	0.54
the jews are gearing up to help usher in democrats in <number> twitter facebook msm google youtube all upping their censorships of the right pro whites	toxic	non-toxic	0.51
amusing to think that white people will never be given a referendum on expelling the jews from our nations when you think about things in such terms it becomes abundantly clear that we are expected by our governments to suffer jewish parasitism without the right to defend ourselves	toxic	non-toxic	0.50
it all hehe haha until we show up at your local favorite hang out spot and wait for you to come out in the van they go one little heeb at a time whatever it takes	toxic	non-toxic	0.53
<user> <user> jews and the stupid holocaust myth	toxic	non-toxic	0.52
<user> jews will never be on the blockchain because you can not trust them	toxic	non-toxic	0.51
<user> <user> it the jews not europeans stop shifting blame on gentiles	toxic	non-toxic	0.57

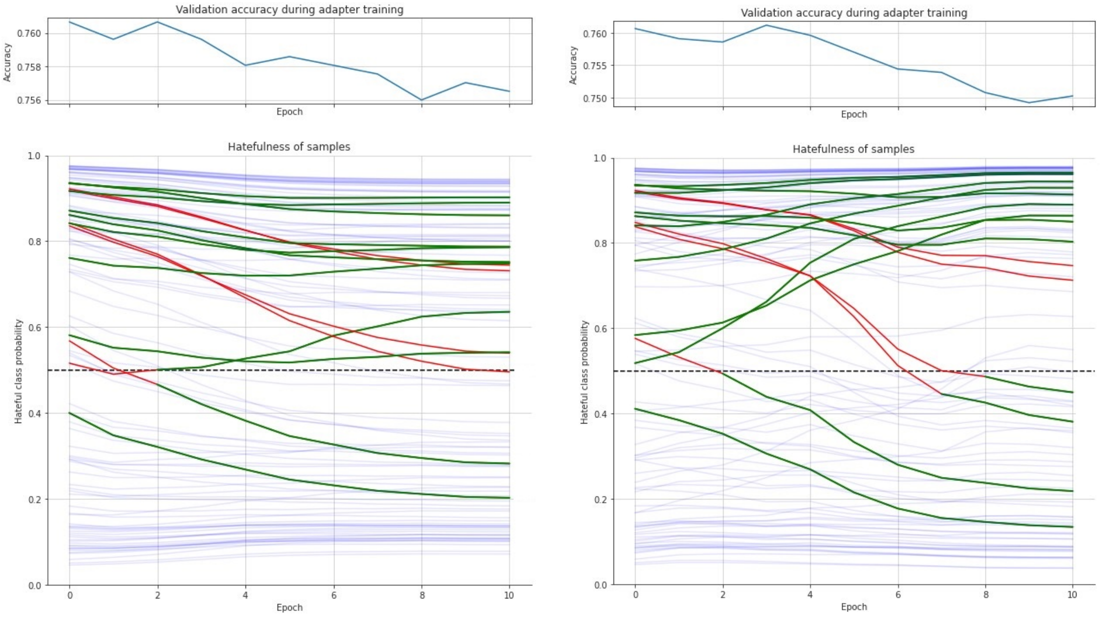
Table 4: Misclassified with the lowest confidence samples used for manual annotation.

Sample	**Original Label**	**Prediction**	**Confidence Score**
it amazing how long <user> talked without giving any answers i learned way more about <user> thoughts and life than sargon i thought only jews could bloviate that hard hm	toxic	non-toxic	0.95
would rather have terrorism over not being able to boycott an allied country this talking point must be important because he repeats it but i can not make out what on earth it supposed to mean if you do not buy stuff from jews they will fake arabs taking down more nyc landmarks	non-toxic	toxic	0.72
<user> <user> um akshually star trek were the space commies and star wars jedis are just space jews	toxic	non-toxic	0.87
<user> apparently not so obvious to some esp the homosexual jewish media that wants women to look like boys and people who think that is traditional rather than feminist	toxic	non-toxic	0.84
how long will jews be a majority in israel	non-toxic	toxic	0.88
sorry i was referring to the holocaust story i believe it to be a lie	toxic	non-toxic	0.89
what do you mean goes and buys their house not only predates the ghetto it was there before ghetto people came to exist why do you want white people to take a stand for their home but jews to flee theirs	non-toxic	toxic	0.77
i swear college is a hot house full of judaised professors with white youth being indoctrinated into this nonsense or have live there days out in fear of being ousted as racists and proud members of white race	toxic	non-toxic	0.90
the jews and muslims grew up together and the jews gave birth to christianity catholicism and later on mormonism like a big religion family	non-toxic	toxic	0.78
calling me a kike is not an insult	non-toxic	toxic	0.97
greg susan frank rodrick and manny heffley are all jews	non-toxic	toxic	0.67
a few months ago i said that anti semitism doesnt exists naming a group of people is a branch of social sciences called sociology well jews are at war with social sciences now you simply cant name them	toxic	non-toxic	0.74

Table 5: Misclassified with the highest confidence samples used for manual annotation.

C.2 Feedback Mechanisms Comparison

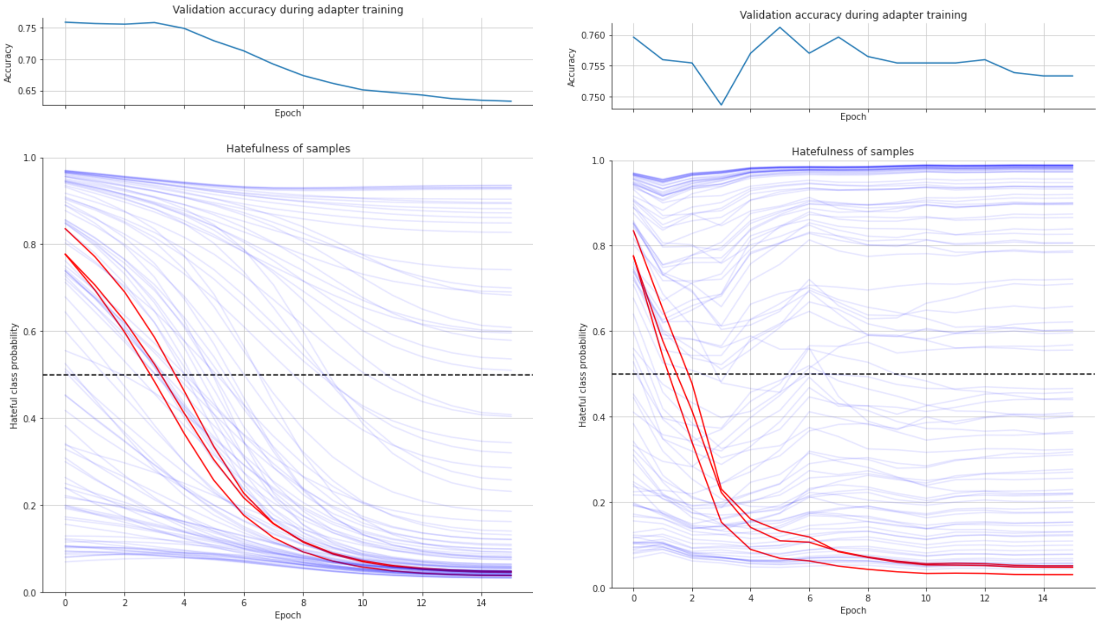
In Section 4, we report the results of the model trained on feedback in two setups: (i) without balancing and (ii) with balancing via using original samples from HateXplain dataset. The comparison between these two setups is visualized in Figures 8 and 9. We tested our approach on the local feedback on the Jewish target group samples as well as samples containing the “Jewish” keyword. For both setups, with balancing, the training procedure runs more stable. The model’s performance on other samples from HateXplain dataset changes slightly and the adjustment of its behavior on the marked-up samples proceeds more rapidly.



(a) Training without feedback balancing.

(b) Training with feedback balancing.

Figure 8: The comparison of training procedure with and without feedback balancing. Here, the results of local feedback on the least confident misclassified samples from the Jewish target group are shown. We can observe that training with a balanced dataset runs more stable without significant influence on the overall model’s domain knowledge.



(a) Training without feedback balancing.

(b) Training with feedback balancing.

Figure 9: The comparison of training procedure with and without feedback balancing. Here, the results of local feedback on misclassified samples with "jewish" keywords are shown. We can observe that training with balanced dataset runs more stable without significant influence on overall model's domain knowledge.

D Supplementary Video Demo

A supplementary video showcase can be found on Youtube[14].

[14] https://www.youtube.com/watch?v=BzzoQzTsrLo

BIBLIOGRAPHY

Abnar, Samira and Willem Zuidema (July 2020). "Quantifying Attention Flow in Transformers." In: *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Online: Association for Computational Linguistics, pp. 4190–4197. doi: 10.18653/v1/2020.acl-main.385. url: https://aclanthology.org/2020.acl-main.385.

Aigrain, Jonathan and Marcin Detyniecki (2019). "Detecting Adversarial Examples and Other Misclassifications in Neural Networks by Introspection." In: *CoRR* abs/1905.09186. arXiv: 1905.09186. url: http://arxiv.org/abs/1905.09186.

Al Kuwatly, Hala, Maximilian Wich, and Georg Groh (Nov. 2020). "Identifying and Measuring Annotator Bias Based on Annotators' Demographic Characteristics." In: *Proceedings of the Fourth Workshop on Online Abuse and Harms*. Online: Association for Computational Linguistics, pp. 184–190. doi: 10.18653/v1/2020.alw-1.21. url: https://aclanthology.org/2020.alw-1.21.

Alzantot, Moustafa, Yash Sharma, Ahmed Elgohary, Bo-Jhang Ho, Mani Srivastava, and Kai-Wei Chang (Oct. 2018). "Generating Natural Language Adversarial Examples." In: *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*. Brussels, Belgium: Association for Computational Linguistics, pp. 2890–2896. doi: 10.18653/v1/D18-1316. url: https://aclanthology.org/D18-1316.

Amershi, Saleema, Maya Cakmak, W. Bradley Knox, and Todd Kulesza (2014). "Power to the People: The Role of Humans in Interactive Machine Learning." In: *AI Mag.* 35.4, pp. 105–120. doi: 10.1609/aimag.v35i4.2513. url: https://doi.org/10.1609/aimag.v35i4.2513.

Arrieta, Alejandro Barredo, Natalia Dıaz-Rodrıguez, Javier Del Ser, Adrien Bennetot, Siham Tabik, Alberto Barbado, Salvador Garcıa, Sergio Gil-López, Daniel Molina, Richard Benjamins, et al. (2020). "Explainable Artificial Intelligence (XAI): Concepts, taxonomies, opportunities and challenges toward responsible AI." In: *Information Fusion* 58, pp. 82–115.

Atanasova, Pepa, Jakob Grue Simonsen, Christina Lioma, and Isabelle Augenstein (Nov. 2020). "A Diagnostic Study of Explainability Techniques for Text Classification." In: *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Online: Association for Computational Linguistics, pp. 3256–3274. DOI: 10.18653/v1/2020.emnlp-main.263. URL: https://aclanthology.org/2020.emnlp-main.263.

Bastings, Jasmijn, Sebastian Ebert, Polina Zablotskaia, Anders Sandholm, and Katja Filippova (2021). "" Will You Find These Shortcuts?" A Protocol for Evaluating the Faithfulness of Input Salience Methods for Text Classification." In: *arXiv preprint arXiv:2111.07367*.

Bastings, Jasmijn and Katja Filippova (Nov. 2020). "The elephant in the interpretability room: Why use attention as explanation when we have saliency methods?" In: *Proceedings of the Third BlackboxNLP Workshop on Analyzing and Interpreting Neural Networks for NLP*. Online: Association for Computational Linguistics, pp. 149–155. DOI: 10.18653/v1/2020.blackboxnlp-1.14. URL: https://aclanthology.org/2020.blackboxnlp-1.14.

Beck, Tilman, Bela Bohlender, Christina Viehmann, Vincent Hane, Yanik Adamson, Jaber Khuri, Jonas Brossmann, Jonas Pfeiffer, and Iryna Gurevych (May 2022). "AdapterHub Playground: Simple and Flexible Few-Shot Learning with Adapters." In: *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*. Dublin, Ireland: Association for Computational Linguistics, pp. 61–75. DOI: 10.18653/v1/2022.acl-demo.6. URL: https://aclanthology.org/2022.acl-demo.6.

Belinkov, Yonatan, Sebastian Gehrmann, and Ellie Pavlick (July 2020). "Interpretability and Analysis in Neural NLP." In: *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: Tutorial Abstracts*. Online: Association for Computational Linguistics, pp. 1–5. DOI: 10.18653/v1/2020.acl-tutorials.1. URL: https://aclanthology.org/2020.acl-tutorials.1.

Belinkov, Yonatan and James Glass (2019). "Analysis Methods in Neural Language Processing: A Survey." In: *Transactions of the Association for Computational Linguistics* 7, pp. 49–72. DOI: 10.1162/tacl_a_00254. URL: https://aclanthology.org/Q19-1004.

Bhatt, Umang, Alice Xiang, Shubham Sharma, Adrian Weller, Ankur Taly, Yunhan Jia, Joydeep Ghosh, Ruchir Puri, José M. F. Moura, and Peter Eckersley (2020). "Explainable Machine Learning in Deployment." In: *Proceedings of the 2020 Conference on Fairness, Accountability, and Transparency*. FAT* '20. Barcelona, Spain: Association for Computing Machinery, pp. 648–657. ISBN: 9781450369367. DOI: 10.1145/3351095.3375624. URL: https://doi.org/10.1145/3351095.3375624.

Bibal, Adrien, Rémi Cardon, David Alfter, Rodrigo Wilkens, Xiaoou Wang, Thomas François, and Patrick Watrin (May 2022). "Is Attention Explanation? An Introduction to the Debate." In: *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Dublin, Ireland: Association for Computational Linguistics, pp. 3889–3900. DOI: 10.18653/v1/2022.acl-long.269. URL: https://aclanthology.org/2022.acl-long.269.

Bolukbasi, Tolga, Kai-Wei Chang, James Y Zou, Venkatesh Saligrama, and Adam T Kalai (2016). "Man is to computer programmer as woman is to homemaker? debiasing word embeddings." In: *Advances in neural information processing systems*, pp. 4349–4357.

Brown, Tom, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. (2020). "Language models are few-shot learners." In: *Advances in neural information processing systems* 33, pp. 1877–1901.

Byrne, Ruth MJ (2019). "Counterfactuals in Explainable Artificial Intelligence (XAI): Evidence from Human Reasoning." In: *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence (IJCAI-19)*.

Camburu, Oana-Maria, Tim Rocktäschel, Thomas Lukasiewicz, and Phil Blunsom (2018). "e-snli: Natural language inference with natural language explanations." In: *Advances in Neural Information Processing Systems* 31.

Chen, Hanjie, Guangtao Zheng, and Yangfeng Ji (July 2020). "Generating Hierarchical Explanations on Text Classification via Feature Interaction Detection." In: *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Online: Association for Computational Linguistics, pp. 5578–5593. DOI: 10.18653/v1/2020.acl-main.494. URL: https://aclanthology.org/2020.acl-main.494.

Chen, Jianbo and Michael Jordan (2020). "Ls-tree: Model interpretation when the data are linguistic." In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 34. 04, pp. 3454–3461.

Chen, Tianqi and Carlos Guestrin (2016). "XGBoost: A Scalable Tree Boosting System." In: *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. KDD '16. Association for Computing Machinery, pp. 785–794.

Covert, Ian, Scott M Lundberg, and Su-In Lee (2020). "Understanding global feature contributions with additive importance measures." In: *Advances in Neural Information Processing Systems* 33, pp. 17212–17223.

Danilevsky, Marina, Kun Qian, Ranit Aharonov, Yannis Katsis, Ban Kawas, and Prithviraj Sen (Dec. 2020). "A Survey of the State of Explainable AI for Natural Language Processing." In: *Proceedings of the 1st Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 10th International Joint Conference on Natural Language Processing*. Suzhou, China: Association for Computational Linguistics, pp. 447–459. URL: https://aclanthology.org/2020.aacl-main.46.

Davidson, Thomas, Dana Warmsley, Michael Macy, and Ingmar Weber (2017). "Automated hate speech detection and the problem of offensive language." In: *Proceedings of the International AAAI Conference on Web and Social Media*. Vol. 11. 1.

Deng, Jia, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei (2009). "Imagenet: A large-scale hierarchical image database." In: *2009 IEEE conference on computer vision and pattern recognition*. Ieee, pp. 248–255.

Devlin, Jacob, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova (June 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). Minneapolis, Minnesota: Association for Computational Linguistics, pp. 4171–4186. DOI: 10.18653/v1/N19-1423. URL: https://aclanthology.org/N19-1423.

Doshi-Velez, Finale and Been Kim (2017). "Towards a rigorous science of interpretable machine learning." In: *arXiv preprint arXiv:1702.08608*.

Ebrahimi, Javid, Anyi Rao, Daniel Lowd, and Dejing Dou (July 2018). "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)*. Melbourne, Australia: Association for Computational Linguistics, pp. 31–36. DOI: 10.18653/v1/P18-2006. URL: https://aclanthology.org/P18-2006.

Edwards, Lilian and Michael Veale (2017). "Slave to the algorithm: Why a right to an explanation is probably not the remedy you are looking for." In: *Duke L. & Tech. Rev.* 16, p. 18.

Egelman, Serge, Ed H Chi, and Steven Dow (2014). *Crowdsourcing in HCI research*. Springer, pp. 267–289.

Eger, Steffen, Gözde Gül Şahin, Andreas Rücklé, Ji-Ung Lee, Claudia Schulz, Mohsen Mesgar, Krishnkant Swarnkar, Edwin Simpson, and Iryna Gurevych (June 2019). "Text Processing Like Humans Do: Visually Attacking and Shielding NLP Systems." 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)*. Minneapolis, Minnesota: Association for Computational Linguistics, pp. 1634–1647. DOI: 10.18653/v1/N19-1165. URL: https://aclanthology.org/N19-1165.

Eichstaedt, Johannes C, Margaret L Kern, David B Yaden, HA Schwartz, Salvatore Giorgi, Gregory Park, Courtney A Hagan, Victoria A Tobolsky, Laura K Smith, Anneke Buffone, et al. (2021). "Closed-and open-vocabulary approaches to text analysis: A review, quantitative comparison, and recommendations." In: *Psychological Methods* 26.4, p. 398.

Etmann, Christian, Sebastian Lunz, Peter Maass, and Carola Schönlieb (2019). "On the Connection Between Adversarial Robustness and Saliency Map Interpretability." In: *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA*. Ed. by Kamalika Chaudhuri and Ruslan

Salakhutdinov. Vol. 97. Proceedings of Machine Learning Research. PMLR, pp. 1823–1832. URL: http://proceedings.mlr.press/v97/etmann19a.html.

European Commission (2019). "Communication from the Commission to the European Parliament, the Council, the European Economic and Social Committee and the Committee of the Regions. Building Trust in Human-Centric Artificial Intelligence." In: *Com (2019) 168 Final*. URL: https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=COM:2019:0168:FIN.

– (2020). "White Paper on Artificial Intelligence: a European approach to excellence and trust." In: *Com (2020) 65 Final*. URL: https://commission.europa.eu/publications/white-paper-artificial-intelligence-european-approach-excellence-and-trust_en.

European Parliament (2016). "Regulation (EU) 2016/679 of the European Parliament and of the Council of 27 April 2016 on the protection of natural persons with regard to the processing of personal data and on the free movement of such data, and repealing Directive 95/46." In: *Official Journal of the European Union (OJ)* 59.1-88, p. 294.

Fehn Unsvåg, Elise and Björn Gambäck (Oct. 2018). "The Effects of User Features on Twitter Hate Speech Detection." In: *Proceedings of the 2nd Workshop on Abusive Language Online (ALW2)*. Brussels, Belgium: Association for Computational Linguistics, pp. 75–85. DOI: 10.18653/v1/W18-5110. URL: https://aclanthology.org/W18-5110.

Fidel, G., R. Bitton, and A. Shabtai (2020). "When Explainability Meets Adversarial Learning: Detecting Adversarial Examples using SHAP Signatures." In: *2020 International Joint Conference on Neural Networks (IJCNN)*, pp. 1–8.

Galán-Garcıa, Patxi, José Gaviria de la Puerta, Carlos Laorden Gómez, Igor Santos, and Pablo Garcıa Bringas (2016). "Supervised machine learning for the detection of troll profiles in twitter social network: Application to a real case of cyberbullying." In: *Logic Journal of the IGPL* 24.1, pp. 42–53.

Gao, Ji, Jack Lanchantin, Mary Lou Soffa, and Yanjun Qi (2018). "Black-Box Generation of Adversarial Text Sequences to Evade Deep Learning Classifiers." In: *2018 IEEE Security and Privacy Workshops (SPW)*, pp. 50–56.

Garg, Siddhant and Goutham Ramakrishnan (Nov. 2020). "BAE: BERT-based Adversarial Examples for Text Classification." In: *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Online: Association for Computational Linguistics, pp. 6174–6181. DOI: 10.18653/v1/2020.emnlp-main.498. URL: https://aclanthology.org/2020.emnlp-main.498.

Gencheva, Pepa, Ivan Koychev, Lluıs Màrquez, Alberto Barrón-Cedeño, and Preslav Nakov (2019). "A Context-Aware Approach for Detecting Check-Worthy Claims in Political Debates." In: *arXiv preprint arXiv:1912.08084*.

Ghorbani, Amirata, James Wexler, James Y Zou, and Been Kim (2019). "Towards automatic concept-based explanations." In: *Advances in Neural Information Processing Systems* 32.

Ghorbani, Amirata and James Zou (2019). "Data shapley: Equitable valuation of data for machine learning." In: *International Conference on Machine Learning*. PMLR, pp. 2242–2251.

Ghorbani, Amirata and James Y. Zou (2020). "Neuron Shapley: Discovering the Responsible Neurons." In: ed. by Hugo Larochelle, Marc'Aurelio Ranzato, Raia Hadsell, Maria-Florina Balcan, and Hsuan-Tien Lin. URL: https://proceedings.neurips.cc/paper/2020/hash/41c542dfe6e4fc3deb251d64cf6ed2e4-Abstract.html.

Gilmartin, Shannon K, Helen L Chen, Mark F Schar, Qu Jin, George Toye, A Harris, Emily Cao, Emanuel Costache, Maximillian Reithmann, and Sheri D Sheppard (2017). "Designing a longitudinal study of engineering students' innovation and engineering interests and plans: The Engineering Majors Survey Project. EMS 1.0 and 2.0 Technical Report." In: *Stanford University Designing Education Lab, Stanford, CA, Technical Report*.

Goodman, Bryce and Seth Flaxman (2017). "European Union regulations on algorithmic decision-making and a "right to explanation"." In: *AI magazine* 38.3, pp. 50–57.

Grau, Michelle Marie, Sheri Sheppard, Shannon Katherine Gilmartin, and Beth Rieken (2016). "What do you want to do with your life? Insights into how engineering

Students think about their future career plans." In: *2016 ASEE Annual Conference & Exposition*.

Guidotti, Riccardo, Anna Monreale, Salvatore Ruggieri, Franco Turini, Fosca Giannotti, and Dino Pedreschi (2019). "A survey of methods for explaining black box models." In: *ACM computing surveys (CSUR)* 51.5, p. 93.

Hamilton, William L., Zhitao Ying, and Jure Leskovec (2017). "Inductive Representation Learning on Large Graphs." In: *NIPS*, pp. 1024–1034.

Han, Xiaochuang, Byron C. Wallace, and Yulia Tsvetkov (July 2020). "Explaining Black Box Predictions and Unveiling Data Artifacts through Influence Functions." In: *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Online: Association for Computational Linguistics, pp. 5553–5563. DOI: 10.18653/v1/2020.acl-main.492. URL: https://aclanthology.org/2020.acl-main.492.

Hao, Yiding (Nov. 2020). "Evaluating Attribution Methods using White-Box LSTMs." In: *Proceedings of the Third BlackboxNLP Workshop on Analyzing and Interpreting Neural Networks for NLP*. Online: Association for Computational Linguistics, pp. 300–313. DOI: 10.18653/v1/2020.blackboxnlp-1.28. URL: https://aclanthology.org/2020.blackboxnlp-1.28.

Hendrycks, Dan and Kevin Gimpel (2016). "Early methods for detecting adversarial images." In: *arXiv preprint arXiv:1608.00530*.

Herman, Bernease (2017). "The promise and peril of human evaluation for model interpretability." In: *arXiv preprint arXiv:1711.07414*.

Hochreiter, Sepp and Jürgen Schmidhuber (1997). "Long short-term memory." In: *Neural computation* 9.8, pp. 1735–1780.

Houlsby, Neil, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin De Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly (2019). "Parameter-efficient transfer learning for NLP." In: *International Conference on Machine Learning*. PMLR, pp. 2790–2799.

Huber, Lukas, Marc Alexander Kühn, Edoardo Mosca, and Georg Groh (May 2022). "Detecting Word-Level Adversarial Text Attacks via SHapley Additive exPlanations." In: *Proceedings of the 7th Workshop on Representation Learning for NLP*. Dublin, Ireland: Association for Computational Linguistics, pp. 156–166. DOI: 10.18653/v1/2022.repl4nlp-1.16. URL: https://aclanthology.org/2022.repl4nlp-1.16.

Ignat, Oana, Zhijing Jin, Artem Abzaliev, Laura Biester, Santiago Castro, Naihao Deng, Xinyi Gao, Aylin Gunal, Jacky He, Ashkan Kazemi, et al. (2023). "A PhD Student's Perspective on Research in NLP in the Era of Very Large Language Models." In: *arXiv preprint arXiv:2305.12544*.

Jacovi, Alon and Yoav Goldberg (July 2020). "Towards Faithfully Interpretable NLP Systems: How Should We Define and Evaluate Faithfulness?" In: *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Online: Association for Computational Linguistics, pp. 4198–4205. DOI: 10.18653/v1/2020.acl-main.386. URL: https://aclanthology.org/2020.acl-main.386.

Jacovi, Alon, Ana Marasović, Tim Miller, and Yoav Goldberg (2021). "Formalizing trust in artificial intelligence: Prerequisites, causes and goals of human trust in ai." In: *Proceedings of the 2021 ACM conference on fairness, accountability, and transparency*, pp. 624–635.

Jain, Sarthak and Byron C Wallace (2019). "Attention is not explanation." In: *arXiv preprint arXiv:1902.10186*.

Japkowicz, Nathalie and Mohak Shah (2011). *Evaluating learning algorithms: a classification perspective*. Cambridge University Press.

Ji, Ziwei, Nayeon Lee, Rita Frieske, Tiezheng Yu, Dan Su, Yan Xu, Etsuko Ishii, Yejin Bang, Andrea Madotto, and Pascale Fung (2022). "Survey of hallucination in natural language generation." In: *ACM Computing Surveys*.

Kim, Been, Caleb M Chacha, and Julie A Shah (2015). "Inferring team task plans from human meetings: A generative modeling approach with logic-based prior." In: *Journal of Artificial Intelligence Research* 52, pp. 361–398.

Kim, Been, Cynthia Rudin, and Julie A Shah (2014). "The bayesian case model: A generative approach for case-based reasoning and prototype classification." In: *Advances in Neural Information Processing Systems*, pp. 1952–1960.

Kim, Been, Martin Wattenberg, Justin Gilmer, Carrie Cai, James Wexler, Fernanda Viegas, et al. (2018). "Interpretability beyond feature attribution: Quantitative testing with concept activation vectors (tcav).” In: *International conference on machine learning*. PMLR, pp. 2668–2677.

Kitaev, Nikita and Dan Klein (July 2018). “Constituency Parsing with a Self-Attentive Encoder.” In: *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Melbourne, Australia: Association for Computational Linguistics, pp. 2676–2686. DOI: 10.18653/v1/P18-1249. URL: https://aclanthology.org/P18-1249.

Knapič, Samanta, Avleen Malhi, Rohit Saluja, and Kary Främling (2021). “Explainable artificial intelligence for human decision support system in the medical domain.” In: *Machine Learning and Knowledge Extraction* 3.3, pp. 740–770.

Koh, Pang Wei and Percy Liang (2017). “Understanding black-box predictions via influence functions.” In: *International conference on machine learning*. PMLR, pp. 1885–1894.

Kulesza, Todd, Weng-Keen Wong, Simone Stumpf, Stephen Perona, Rachel White, Margaret M. Burnett, Ian Oberst, and Amy J. Ko (2009). “Fixing the program my computer learned: barriers for end users, challenges for the machine.” In: *Proceedings of the 14th International Conference on Intelligent User Interfaces, IUI 2009, Sanibel Island, Florida, USA, February 8-11, 2009*. Ed. by Cristina Conati, Mathias Bauer, Nuria Oliver, and Daniel S. Weld. ACM, pp. 187–196. DOI: 10.1145/1502650.1502678. URL: https://doi.org/10.1145/1502650.1502678.

Kumar, I Elizabeth, Suresh Venkatasubramanian, Carlos Scheidegger, and Sorelle Friedler (2020). “Problems with Shapley-value-based explanations as feature importance measures.” In: *International Conference on Machine Learning*. PMLR, pp. 5491–5500.

Kumar, Sawan and Partha Talukdar (July 2020). “NILE : Natural Language Inference with Faithful Natural Language Explanations.” In: *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Online: Association for Computational Linguistics, pp. 8730–8742. DOI: 10.18653/v1/2020.acl-main.771. URL: https://aclanthology.org/2020.acl-main.771.

Lai, Vivian, Han Liu, and Chenhao Tan (2020). “" Why is' Chicago'deceptive?" Towards Building Model-Driven Tutorials for Humans.” In: *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*, pp. 1–13.

Lei, Qi, Lingfei Wu, Pin-Yu Chen, Alex Dimakis, Inderjit S Dhillon, and Michael J Witbrock (2019). “Discrete adversarial attacks and submodular optimization with applications to text classification.” In: *Proceedings of Machine Learning and Systems* 1, pp. 146–165.

Lertvittayakumjorn, Piyawat, Ivan Petej, Yang Gao, Yamuna Krishnamurthy, Anna Van Der Gaag, Robert Jago, and Kostas Stathis (Aug. 2021). “Supporting Complaints Investigation for Nursing and Midwifery Regulatory Agencies.” In: *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing: System Demonstrations*. Online: Association for Computational Linguistics, pp. 81–91. DOI: 10.18653/v1/2021.acl-demo.10. URL: https://aclanthology.org/2021.acl-demo.10.

Lertvittayakumjorn, Piyawat, Lucia Specia, and Francesca Toni (Nov. 2020). “FIND: Human-in-the-Loop Debugging Deep Text Classifiers.” In: *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Online: Association for Computational Linguistics, pp. 332–348. DOI: 10.18653/v1/2020.emnlp-main.24. URL: https://aclanthology.org/2020.emnlp-main.24.

Lertvittayakumjorn, Piyawat and Francesca Toni (2021). “Explanation-Based Human Debugging of NLP Models: A Survey.” In: *Transactions of the Association for Computational Linguistics* 9, pp. 1508–1528. DOI: 10.1162/tacl_a_00440. URL: https://aclanthology.org/2021.tacl-1.90.

Li, Jiwei, Xinlei Chen, Eduard Hovy, and Dan Jurafsky (2016). “Visualizing and Understanding Neural Models in NLP.” In: *Proceedings of NAACL-HLT*, pp. 681–691.

Lipton, Peter (1990). “Contrastive explanation.” In: *Royal Institute of Philosophy Supplements* 27, pp. 247–266.

Lipton, Zachary C (2016). “The mythos of model interpretability (2016).” In: *arXiv preprint arXiv:1606.03490*.

Liu, Hui, Qingyu Yin, and William Yang Wang (July 2019). “Towards Explainable NLP: A Generative Explanation Framework for Text Classification.” In: *Proceedings of the*

57th Annual Meeting of the Association for Computational Linguistics. Florence, Italy: Association for Computational Linguistics, pp. 5570–5581. DOI: 10.18653/v1/P19-1560. URL: https://aclanthology.org/P19-1560.

Liu, Pengfei, Jinlan Fu, Yang Xiao, Weizhe Yuan, Shuaichen Chang, Junqi Dai, Yixin Liu, Zihuiwen Ye, and Graham Neubig (Aug. 2021). "ExplainaBoard: An Explainable Leaderboard for NLP." In: *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing: System Demonstrations*. Online: Association for Computational Linguistics, pp. 280–289. DOI: 10.18653/v1/2021.acl-demo.34. URL: https://aclanthology.org/2021.acl-demo.34.

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

Locke, Saskia, Anthony Bashall, Sarah Al-Adely, John Moore, Anthony Wilson, and Gareth B Kitchen (2021). "Natural language processing in medicine: a review." In: *Trends in Anaesthesia and Critical Care* 38, pp. 4–9.

Lundberg, Scott M, Gabriel Erion, Hugh Chen, Alex DeGrave, Jordan M Prutkin, Bala Nair, Ronit Katz, Jonathan Himmelfarb, Nisha Bansal, and Su-In Lee (2020). "From local explanations to global understanding with explainable AI for trees." In: *Nature machine intelligence* 2.1, pp. 56–67.

Lundberg, Scott M, Gabriel G Erion, and Su-In Lee (2018). "Consistent individualized feature attribution for tree ensembles." In: *arXiv preprint arXiv:1802.03888*.

Lundberg, Scott M and Su-In Lee (2017). "A unified approach to interpreting model predictions." In: *Advances in Neural Information Processing Systems*, pp. 4765–4774.

Lundervold, Alexander Selvikvåg and Arvid Lundervold (2019). "An overview of deep learning in medical imaging focusing on MRI." In: *Zeitschrift für Medizinische Physik* 29.2, pp. 102–127.

Madsen, Andreas, Siva Reddy, and Sarath Chandar (Dec. 2022). "Post-Hoc Interpretability for Neural NLP: A Survey." In: *ACM Comput. Surv.* 55.8. ISSN: 0360-0300. DOI: 10.1145/3546577. URL: https://doi.org/10.1145/3546577.

Marques, Max RS, Tommaso Bianco, Maxime Roodnejad, Thomas Baduel, and Claude Berrou (2019). "Machine learning for explaining and ranking the most influential matters of law." In: *Proceedings of the Seventeenth International Conference on Artificial Intelligence and Law*, pp. 239–243.

Mathew, Binny, Punyajoy Saha, Seid Muhie Yimam, Chris Biemann, Pawan Goyal, and Animesh Mukherjee (2021). "HateXplain: A Benchmark Dataset for Explainable Hate Speech Detection." In: *Thirty-Fifth AAAI Conference on Artificial Intelligence, AAAI 2021, Thirty-Third Conference on Innovative Applications of Artificial Intelligence, IAAI 2021, The Eleventh Symposium on Educational Advances in Artificial Intelligence, EAAI 2021, Virtual Event, February 2-9, 2021*. AAAI Press, pp. 14867–14875. URL: https://ojs.aaai.org/index.php/AAAI/article/view/17745.

McInnes, Leland, John Healy, and James Melville (2020). *UMAP: Uniform Manifold Approximation and Projection for Dimension Reduction*. arXiv: 1802.03426 [stat.ML].

Merrick, Luke and Ankur Taly (2020). "The explanation game: Explaining machine learning models using shapley values." In: *International Cross-Domain Conference for Machine Learning and Knowledge Extraction*. Springer, pp. 17–38.

Miller, Tim (2019). "Explanation in artificial intelligence: Insights from the social sciences." In: *Artificial Intelligence* 267, pp. 1–38.

Min, Bonan, Hayley Ross, Elior Sulem, Amir Pouran Ben Veyseh, Thien Huu Nguyen, Oscar Sainz, Eneko Agirre, Ilana Heinz, and Dan Roth (2021). "Recent advances in natural language processing via large pre-trained language models: A survey." In: *arXiv preprint arXiv:2111.01243*.

Mishra, Pushkar, Marco Del Tredici, Helen Yannakoudakis, and Ekaterina Shutova (Aug. 2018). "Author Profiling for Abuse Detection." In: *Proceedings of the 27th International Conference on Computational Linguistics*. Santa Fe, New Mexico, USA: Association for Computational Linguistics, pp. 1088–1098. URL: https://aclanthology.org/C18-1093.

– (June 2019). "Abusive Language Detection with Graph Convolutional Networks." 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 Pa-*

pers). Minneapolis, Minnesota: Association for Computational Linguistics, pp. 2145–2150. DOI: 10.18653/v1/N19-1221. URL: https://aclanthology.org/N19-1221.

Mittelstadt, Brent, Chris Russell, and Sandra Wachter (2019). "Explaining explanations in AI." In: *Proceedings of the conference on fairness, accountability, and transparency*, pp. 279–288.

Mohseni, Sina and Eric D. Ragan (2018). "A Human-Grounded Evaluation Benchmark for Local Explanations of Machine Learning." In: *arXiv preprint arXiv:1801.05075*.

Molnar, Christoph (2019). *Interpretable Machine Learning. A Guide for Making Black Box Models Explainable*. https://christophm.github.io/interpretable-ml-book/.

Monarch, Robert Munro (2021). *Human-in-the-Loop Machine Learning: Active learning and annotation for human-centered AI*. Simon and Schuster.

Mosca, Edoardo (2020). "Explainability of hate speech detection models (Master Thesis)." In: *Technical University of Munich, Department of Mathematics*. URL: https://soc.cit.tum.de/persons/edoardo-mosca/Master_Thesis.pdf.

Mosca, Edoardo, Shreyash Agarwal, Javier Rando Ramırez, and Georg Groh (May 2022). ""That Is a Suspicious Reaction!": Interpreting Logits Variation to Detect NLP Adversarial Attacks." In: *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Dublin, Ireland: Association for Computational Linguistics, pp. 7806–7816. DOI: 10.18653/v1/2022.acl-long.538. URL: https://aclanthology.org/2022.acl-long.538.

Mosca, Edoardo, Daryna Dementieva, Tohid Ebrahim Ajdari, Maximilian Kummeth, Kirill Gringauz, and Georg Groh (2023). "IFAN: An Explainability-Focused Interaction Framework for Humans and NLP Models." In: *arXiv preprint arXiv:2303.03124*. (Accepted at AACL, Nov 2023). URL: https://arxiv.org/abs/2303.03124.

Mosca, Edoardo, Defne Demirtürk, Luca Mülln, Fabio Raffagnato, and Georg Groh (May 2022). "GrammarSHAP: An Efficient Model-Agnostic and Structure-Aware NLP Explainer." In: *Proceedings of the First Workshop on Learning with Natural Language Supervision*. Dublin, Ireland: Association for Computational Linguistics, pp. 10–16. DOI: 10.18653/v1/2022.lnls-1.2. URL: https://aclanthology.org/2022.lnls-1.2.

Mosca, Edoardo, Katharina Hermann, Tobias Eder, and Georg Groh (July 2022). "Explaining Neural NLP Models for the Joint Analysis of Open-and-Closed-Ended Survey Answers." In: *Proceedings of the 2nd Workshop on Trustworthy Natural Language Processing (TrustNLP 2022)*. Seattle, U.S.A.: Association for Computational Linguistics, pp. 49–63. DOI: 10.18653/v1/2022.trustnlp-1.5. URL: https://aclanthology.org/2022.trustnlp-1.5.

Mosca, Edoardo, Ferenc Szigeti, Stella Tragianni, Daniel Gallagher, and Georg Groh (Oct. 2022). "SHAP-Based Explanation Methods: A Review for NLP Interpretability." In: *Proceedings of the 29th International Conference on Computational Linguistics*. Gyeongju, Republic of Korea: International Committee on Computational Linguistics, pp. 4593–4603. URL: https://aclanthology.org/2022.coling-1.406.

Mosca, Edoardo, Maximilian Wich, and Georg Groh (June 2021). "Understanding and Interpreting the Impact of User Context in Hate Speech Detection." In: *Proceedings of the Ninth International Workshop on Natural Language Processing for Social Media*. Online: Association for Computational Linguistics, pp. 91–102. DOI: 10.18653/v1/2021.socialnlp-1.8. URL: https://aclanthology.org/2021.socialnlp-1.8.

Mozes, Maximilian, Pontus Stenetorp, Bennett Kleinberg, and Lewis Griffin (Apr. 2021). "Frequency-Guided Word Substitutions for Detecting Textual Adversarial Examples." In: *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*. Online: Association for Computational Linguistics, pp. 171–186. DOI: 10.18653/v1/2021.eacl-main.13. URL: https://aclanthology.org/2021.eacl-main.13.

Murdoch, W. James, Chandan Singh, Karl Kumbier, Reza Abbasi-Asl, and Bin Yu (2019). "Definitions, methods, and applications in interpretable machine learning." In: *Proceedings of the National Academy of Sciences* 116.44, pp. 22071–22080.

Nagahisarchoghaei, Mohammad, Nasheen Nur, Logan Cummins, Nashtarin Nur, Mirhossein Mousavi Karimi, Shreya Nandanwar, Siddhartha Bhattacharyya, and Shahram Rahimi (2023). "An Empirical Survey on Explainable AI Technologies: Recent Trends, Use-Cases, and Categories from Technical and Application Perspectives." In: *Electronics* 12.5. ISSN: 2079-9292. URL: https://www.mdpi.com/2079-9292/12/5/1092.

Nickerson, Raymond S (1998). “Confirmation bias: A ubiquitous phenomenon in many guises.” In: *Review of general psychology* 2.2, pp. 175–220.

OpenAI (2022). “ChatGPT: Optimizing Language Models for Dialogue.” In: URL: https://openai.com/blog/chatgpt/.

Poerner, Nina, Hinrich Schütze, and Benjamin Roth (July 2018). “Evaluating neural network explanation methods using hybrid documents and morphosyntactic agreement.” In: *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Melbourne, Australia: Association for Computational Linguistics, pp. 340–350. DOI: 10.18653/v1/P18-1032. URL: https://aclanthology.org/P18-1032.

Pruthi, Danish, Bhuwan Dhingra, and Zachary C. Lipton (July 2019). “Combating Adversarial Misspellings with Robust Word Recognition.” In: *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. Florence, Italy: Association for Computational Linguistics, pp. 5582–5591. DOI: 10.18653/v1/P19-1561. URL: https://aclanthology.org/P19-1561.

Pruthi, Garima, Frederick Liu, Satyen Kale, and Mukund Sundararajan (2020). “Estimating training data influence by tracing gradient descent.” In: *Advances in Neural Information Processing Systems* 33, pp. 19920–19930.

Radford, Alec, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever (2018). “Improving Language Understanding by Generative Pre-Training.” In: *OpenAI*. URL: https://openai.com/blog/language-unsupervised/.

Rajani, Nazneen Fatema, Bryan McCann, Caiming Xiong, and Richard Socher (July 2019). “Explain Yourself! Leveraging Language Models for Commonsense Reasoning.” In: *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. Florence, Italy: Association for Computational Linguistics, pp. 4932–4942. DOI: 10.18653/v1/P19-1487. URL: https://aclanthology.org/P19-1487.

Ras, Gabriëlle, Marcel van Gerven, and Pim Haselager (2018). “Explanation methods in deep learning: Users, values, concerns and challenges.” In: *Explainable and Interpretable Models in Computer Vision and Machine Learning*. Springer, pp. 19–36.

Ray, Arijit, Yi Yao, Rakesh Kumar, Ajay Divakaran, and Giedrius Burachas (2019). “Can You Explain That? Lucid Explanations Help Human-AI Collaborative Image Retrieval." In: *Proceedings of the AAAI Conference on Human Computation and Crowdsourcing*. Vol. 7. 1, pp. 153–161.

Ren, Shuhuai, Yihe Deng, Kun He, and Wanxiang Che (July 2019). "Generating Natural Language Adversarial Examples through Probability Weighted Word Saliency." In: *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. Florence, Italy: Association for Computational Linguistics, pp. 1085–1097. DOI: 10.18653/v1/P19-1103. URL: https://aclanthology.org/P19-1103.

Ribeiro, Manoel Horta, Pedro H Calais, Yuri A Santos, Virgılio AF Almeida, and Wagner Meira Jr (2018). "Characterizing and detecting hateful users on twitter." In: *Twelfth international AAAI conference on web and social media*.

Ribeiro, Marco Tulio and Scott Lundberg (May 2022). "Adaptive Testing and Debugging of NLP Models." In: *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Dublin, Ireland: Association for Computational Linguistics, pp. 3253–3267. DOI: 10.18653/v1/2022.acl-long.230. URL: https://aclanthology.org/2022.acl-long.230.

Ribeiro, Marco Tulio, Sameer Singh, and Carlos Guestrin (2016). "Why should i trust you?: Explaining the predictions of any classifier." In: *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, pp. 1135–1144.

– (July 2018). "Semantically Equivalent Adversarial Rules for Debugging NLP models." In: *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Melbourne, Australia: Association for Computational Linguistics, pp. 856–865. DOI: 10.18653/v1/P18-1079. URL: https://aclanthology.org/P18-1079.

Ribera, Mireia and Agata Lapedriza (2019). "Can we do better explanations? A proposal of user-centered explainable AI." In: *IUI Workshops*.

Ross, Alexis, Ana Marasović, and Matthew Peters (Aug. 2021). "Explaining NLP Models via Minimal Contrastive Editing (MiCE)." In: *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*. Online: Association for Computational Linguistics, pp. 3840–3852. DOI: 10.18653/v1/2021.findings-acl.336. URL: https://aclanthology.org/2021.findings-acl.336.

Rudin, Cynthia and Berk Ustun (2018). “Optimized scoring systems: Toward trust in machine learning for healthcare and criminal justice.” In: *Interfaces* 48.5, pp. 449–466.

Sanh, Victor, Lysandre Debut, Julien Chaumond, and Thomas Wolf (2019). “DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter.” In: *2019 5th Workshop on Energy Efficient Machine Learning and Cognitive Computing - NeurIPS 2019*.

Scao, Teven Le et al. (2022). “BLOOM: A 176B-Parameter Open-Access Multilingual Language Model.” In: *CoRR* abs/2211.05100. DOI: 10.48550/arXiv.2211.05100. arXiv: 2211.05100. URL: https://doi.org/10.48550/arXiv.2211.05100.

Schuster, M. and K. K. Paliwal (1997). “Bidirectional recurrent neural networks.” In: *IEEE Transactions on Signal Processing* 45.11, pp. 2673–2681. DOI: 10.1109/78.650093.

Selvaraju, Ramprasaath R, Stefan Lee, Yilin Shen, Hongxia Jin, Shalini Ghosh, Larry Heck, Dhruv Batra, and Devi Parikh (2019). “Taking a hint: Leveraging explanations to make vision and language models more grounded.” In: *Proceedings of the IEEE International Conference on Computer Vision*, pp. 2591–2600.

Serrano, Sofia and Noah A Smith (2019). “Is Attention Interpretable?” In: *arXiv preprint arXiv:1906.03731*.

Shapley, Lloyd S (1953). “A value for n-person games.” In: *Contributions to the Theory of Games 2.28*, pp. 307–317.

Shrikumar, Avanti, Peyton Greenside, and Anshul Kundaje (2017). “Learning important features through propagating activation differences.” In: *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pp. 3145–3153.

Shukla, Abhay, Paheli Bhattacharya, Soham Poddar, Rajdeep Mukherjee, Kripabandhu Ghosh, Pawan Goyal, and Saptarshi Ghosh (Nov. 2022). “Legal Case Document Summarization: Extractive and Abstractive Methods and their Evaluation.” In: *Proceedings of the 2nd Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 12th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*. Online only: Association for Computational Linguistics, pp. 1048–1064. URL: https://aclanthology.org/2022.aacl-main.77.

Singh, Chandan, W James Murdoch, and Bin Yu (2018). "Hierarchical interpretations for neural network predictions." In: *International Conference on Learning Representations*.

Smith-Renner, Alison, Ron Fan, Melissa Birchfield, Tongshuang Wu, Jordan L. Boyd-Graber, Daniel S. Weld, and Leah Findlater (2020). "No Explainability without Accountability: An Empirical Study of Explanations and Feedback in Interactive ML." In: *CHI '20: CHI Conference on Human Factors in Computing Systems, Honolulu, HI, USA, April 25-30, 2020*. Ed. by Regina Bernhaupt et al. ACM, pp. 1–13. DOI: 10.1145/3313831.3376624. URL: https://doi.org/10.1145/3313831.3376624.

Strout, Julia, Ye Zhang, and Raymond Mooney (Aug. 2019). "Do Human Rationales Improve Machine Explanations?" In: *Proceedings of the 2019 ACL Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*. Florence, Italy: Association for Computational Linguistics, pp. 56–62. DOI: 10.18653/v1/W19-4807. URL: https://aclanthology.org/W19-4807.

Sun, Xiaofei, Diyi Yang, Xiaoya Li, Tianwei Zhang, Yuxian Meng, Han Qiu, Guoyin Wang, Eduard Hovy, and Jiwei Li (2021). "Interpreting deep learning models in natural language processing: A review." In: *arXiv preprint arXiv:2110.10470*.

Sundararajan, Mukund and Amir Najmi (2020). "The many Shapley values for model explanation." In: *International Conference on Machine Learning*. PMLR, pp. 9269–9278.

Sundararajan, Mukund, Ankur Taly, and Qiqi Yan (2017). "Axiomatic attribution for deep networks." In: *Proceedings of the 34th International Conference on Machine Learning-Volume 70*. JMLR. org, pp. 3319–3328.

Sundermann, Camila Vaccari, Marcos Aurélio Domingues, Roberta Akemi Sinoara, Ricardo Marcondes Marcacini, and Solange Oliveira Rezende (2019). "Using Opinion Mining in Context-Aware Recommender Systems: A Systematic Review." In: *Information* 10.2. ISSN: 2078-2489. DOI: 10.3390/info10020042. URL: https://www.mdpi.com/2078-2489/10/2/42.

Szegedy, Christian, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus (2014). "Intriguing properties of neural networks." In: 2nd International Conference on Learning Representations, ICLR 2014.

Tao, Guanhong, Shiqing Ma, Yingqi Liu, and Xiangyu Zhang (2018). “Attacks Meet Interpretability: Attribute-Steered Detection of Adversarial Samples.” In: *Proceedings of the 32nd International Conference on Neural Information Processing Systems*. NIPS’18. Curran Associates Inc., pp. 7728–7739.

Tenney, Ian et al. (Oct. 2020). “The Language Interpretability Tool: Extensible, Interactive Visualizations and Analysis for NLP Models.” In: *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*. Online: Association for Computational Linguistics, pp. 107–118. DOI: 10.18653/v1/2020.emnlp-demos.15. URL: https://aclanthology.org/2020.emnlp-demos.15.

Teso, Stefano and Kristian Kersting (2019). “Explanatory Interactive Machine Learning.” In: *Proceedings of the 2019 AAAI/ACM Conference on AI, Ethics, and Society, AIES 2019, Honolulu, HI, USA, January 27-28, 2019*. Ed. by Vincent Conitzer, Gillian K. Hadfield, and Shannon Vallor. ACM, pp. 239–245. DOI: 10.1145/3306618.3314293. URL: https://doi.org/10.1145/3306618.3314293.

Tsang, Michael, Youbang Sun, Dongxu Ren, and Yan Liu (2018). “Can I trust you more? Model-agnostic hierarchical explanations.” In: *arXiv preprint arXiv:1812.04801*.

Tsipras, Dimitris, Shibani Santurkar, Logan Engstrom, Alexander Turner, and Aleksander Madry (2019). “Robustness May Be at Odds with Accuracy.” In: *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net. URL: https://openreview.net/forum?id=SyxAb30cY7.

Vaswani, Ashish, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin (2017). “Attention is all you need.” In: *Advances in neural information processing systems*, pp. 5998–6008.

Vig, Jesse, Sebastian Gehrmann, Yonatan Belinkov, Sharon Qian, Daniel Nevo, Yaron Singer, and Stuart Shieber (2020). “Investigating gender bias in language models using causal mediation analysis.” In: *Advances in Neural Information Processing Systems* 33, pp. 12388–12401.

Wachter, Sandra, Brent Mittelstadt, and Luciano Floridi (2017). “Why a right to explanation of automated decision-making does not exist in the general data protection regulation.” In: *International Data Privacy Law* 7.2, pp. 76–99.

Wachter, Sandra, Brent Mittelstadt, and Chris Russell (2017). "Counterfactual explanations without opening the black box: Automated decisions and the GDPR." In: *Harv. JL & Tech.* 31, p. 841.

Wallace, Eric, Matt Gardner, and Sameer Singh (Nov. 2020). "Interpreting Predictions of NLP Models." In: *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: Tutorial Abstracts*. Online: Association for Computational Linguistics, pp. 20–23. DOI: 10.18653/v1/2020.emnlp-tutorials.3. URL: https://aclanthology.org/2020.emnlp-tutorials.3.

Wallace, Eric, Jens Tuyls, Junlin Wang, Sanjay Subramanian, Matt Gardner, and Sameer Singh (Nov. 2019). "AllenNLP Interpret: A Framework for Explaining Predictions of NLP Models." In: *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP): System Demonstrations*. Hong Kong, China: Association for Computational Linguistics, pp. 7–12. DOI: 10.18653/v1/D19-3002. URL: https://aclanthology.org/D19-3002.

Wang, Wenqi, Run Wang, Lina Wang, Zhibo Wang, and Aoshuang Ye (2019). "Towards a robust deep neural network in texts: A survey." In: *arXiv preprint arXiv:1902.07285*.

Wang, Xuezhi, Haohan Wang, and Diyi Yang (July 2022). "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*. Seattle, United States: Association for Computational Linguistics, pp. 4569–4586. DOI: 10.18653/v1/2022.naacl-main.339. URL: https://aclanthology.org/2022.naacl-main.339.

Wang, Yaopeng, Lehui Xie, Ximeng Liu, Jia-Li Yin, and Tingjie Zheng (2021). "Model-Agnostic Adversarial Example Detection Through Logit Distribution Learning." In: *2021 IEEE International Conference on Image Processing (ICIP)*, pp. 3617–3621. DOI: 10.1109/ICIP42928.2021.9506292.

Wang, Yuqing, Yun Zhao, and Linda Petzold (2023). "Are Large Language Models Ready for Healthcare? A Comparative Study on Clinical Language Understanding." In: *arXiv preprint arXiv:2304.05368*.

Wang, Zijie J., Dongjin Choi, Shenyu Xu, and Diyi Yang (Apr. 2021). "Putting Humans in the Natural Language Processing Loop: A Survey." In: *Proceedings of the First Workshop on Bridging Human–Computer Interaction and Natural Language Processing*. Online: Association for Computational Linguistics, pp. 47–52. URL: https://aclanthology.org/2021.hcinlp-1.8.

Waseem, Zeerak (Nov. 2016). "Are You a Racist or Am I Seeing Things? Annotator Influence on Hate Speech Detection on Twitter." In: *Proceedings of the First Workshop on NLP and Computational Social Science*. Austin, Texas: Association for Computational Linguistics, pp. 138–142. DOI: 10.18653/v1/W16-5618. URL: https://aclanthology.org/W16-5618.

West, Darrell M (2018). *The future of work: Robots, AI, and automation*. Brookings Institution Press.

Wich, Maximilian, Melissa Breitinger, Wienke Strathern, Marlena Naimarevic, Georg Groh, and Jürgen Pfeffer (2021). "Are your Friends also Haters? Identification of Hater Networks on Social Media: Data Paper." In: *Companion Proc. Web Conference 2021*. ACM.

Wich, Maximilian, Edoardo Mosca, Adrian Gorniak, Johannes Hingerl, and Georg Groh (2021). "Explainable abusive language classification leveraging user and network data." In: *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*. Springer, pp. 481–496. URL: https://2021.ecmlpkdd.org/wp-content/uploads/2021/07/sub_663.pdf.

Wiegreffe, Sarah and Yuval Pinter (Nov. 2019). "Attention is not not Explanation." 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)*. Hong Kong, China: Association for Computational Linguistics, pp. 11–20. DOI: 10.18653/v1/D19-1002. URL: https://aclanthology.org/D19-1002.

Wolf, Thomas et al. (Oct. 2020). "Transformers: State-of-the-Art Natural Language Processing." In: *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*. Online: Association for Computational Linguistics, pp. 38–45. DOI: 10.18653/v1/2020.emnlp-demos.6. URL: https://aclanthology.org/2020.emnlp-demos.6.

Wu, Tongshuang, Marco Tulio Ribeiro, Jeffrey Heer, and Daniel Weld (Aug. 2021). “Polyjuice: Generating Counterfactuals for Explaining, Evaluating, and Improving 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)*. Online: Association for Computational Linguistics, pp. 6707–6723. DOI: 10.18653/v1/2021.acl-long.523. URL: https://aclanthology.org/2021.acl-long.523.

Yao, Huihan, Ying Chen, Qinyuan Ye, Xisen Jin, and Xiang Ren (2021). “Refining language models with compositional explanations.” In: *Advances in Neural Information Processing Systems* 34, pp. 8954–8967.

Ye, Dengpan, Chuanxi Chen, Changrui Liu, Hao Wang, and Shunzhi Jiang (2020). “Detection Defense Against Adversarial Attacks with Saliency Map.” In: *arXiv preprint arXiv:2009.02738*.

Yeh, Chih-Kuan, Been Kim, Sercan Arik, Chun-Liang Li, Tomas Pfister, and Pradeep Ravikumar (2020). “On completeness-aware concept-based explanations in deep neural networks.” In: *Advances in Neural Information Processing Systems* 33, pp. 20554–20565.

Yuan, Xiaoyong, Pan He, Qile Zhu, and Xiaolin Li (2019). “Adversarial examples: Attacks and defenses for deep learning.” In: *IEEE transactions on neural networks and learning systems* 30.9, pp. 2805–2824.

Zhang, Wei Emma, Quan Z. Sheng, Ahoud Alhazmi, and Chenliang Li (2020). “Adversarial Attacks on Deep-Learning Models in Natural Language Processing: A Survey.” In: *ACM Trans. Intell. Syst. Technol.* 11.3.

Zhang, Zhuoren (2021). “ResNet-Based Model for Autonomous Vehicles Trajectory Prediction.” In: *2021 IEEE International Conference on Consumer Electronics and Computer Engineering (ICCECE)*, pp. 565–568. DOI: 10.1109/ICCECE51280.2021.9342418.

Zhao, Wayne Xin, Kun Zhou, Junyi Li, Tianyi Tang, Xiaolei Wang, Yupeng Hou, Yingqian Min, Beichen Zhang, Junjie Zhang, Zican Dong, et al. (2023). “A survey of large language models.” In: *arXiv preprint arXiv:2303.18223*.

Zhou, Yichao, Jyun-Yu Jiang, Kai-Wei Chang, and Wei Wang (Nov. 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)*. Hong Kong, China: Association for Computational Linguistics, pp. 4904–4913. DOI: 10.18653/v1/D19-1496. URL: https://www.aclweb.org/anthology/D19-1496.

Zylberajch, Hugo, Piyawat Lertvittayakumjorn, and Francesca Toni (Aug. 2021). "HILDIF: Interactive Debugging of NLI Models Using Influence Functions." In: *Proceedings of the First Workshop on Interactive Learning for Natural Language Processing*. Online: Association for Computational Linguistics, pp. 1–6. DOI: 10.18653/v1/2021.internlp-1.1. URL: https://aclanthology.org/2021.internlp-1.1.

Lizenzen / Reprint Permissions

Explainable AI for the Human-Centric Development of NLP Models

Edoardo Mosca

November 22, 2023

The dissertation ”Explainable AI for the Human-Centric Development of NLP Models” presents eight studies. Here we provide all information regarding their reprint permissions, confirming that such studies can be reused in the context of the dissertation.

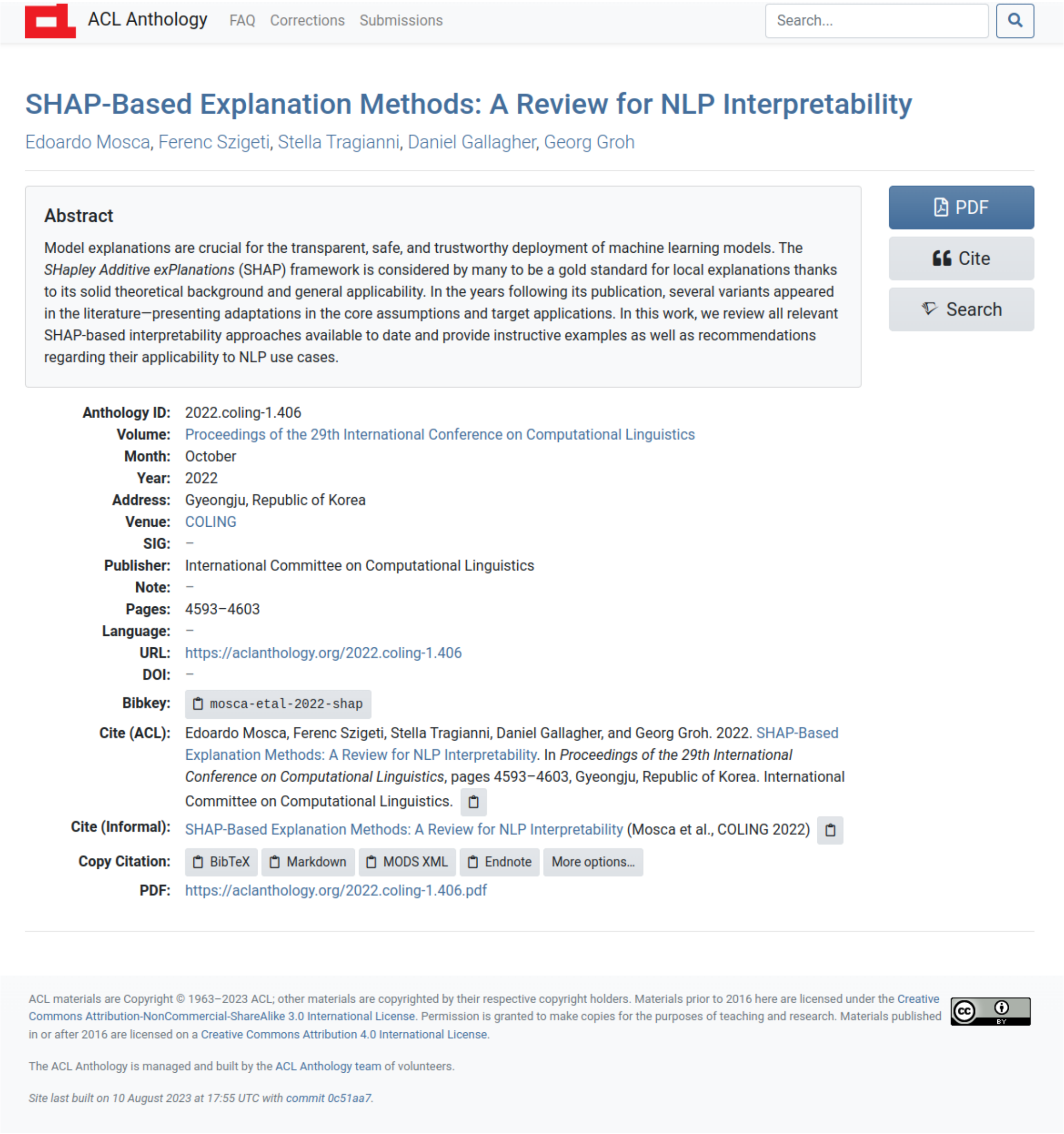
1 Studies

- Study I: Mosca, Szigeti, et al. 2022
- Study II: Mosca, Demirtürk, et al. 2022
- Study III: Mosca, Wich, and Groh 2021
- Study IV: Wich et al. 2021
- Study V: Mosca, Harmann, et al. 2022
- Study VI: Huber et al. 2022
- Study VII: Mosca, Agarwal, et al. 2022
- Study VIII: Mosca, Dementieva, et al. 2023

2 Licences

2.1 Study I

Study I is published through a venue part of the *Association for Computational Linguistics* (ACL) and is public on the ACL Anthology website[1]. All articles in the anthology are published under the Creative Commons 4.0 Deed license (see Figure 1 at the bottom). Such license (CC BY 4.0 Deed) allows copying and redistributing the material in any medium or format for any purpose, even commercially. We attach the corresponding .pdf, retrieved from the website https://creativecommons.org/licenses/by/4.0/.



SHAP-Based Explanation Methods: A Review for NLP Interpretability

Edoardo Mosca, Ferenc Szigeti, Stella Tragianni, Daniel Gallagher, Georg Groh

Abstract

Model explanations are crucial for the transparent, safe, and trustworthy deployment of machine learning models. The *SHapley Additive exPlanations* (SHAP) framework is considered by many to be a gold standard for local explanations thanks to its solid theoretical background and general applicability. In the years following its publication, several variants appeared in the literature—presenting adaptations in the core assumptions and target applications. In this work, we review all relevant SHAP-based interpretability approaches available to date and provide instructive examples as well as recommendations regarding their applicability to NLP use cases.

Anthology ID: 2022.coling-1.406
Volume: Proceedings of the 29th International Conference on Computational Linguistics
Month: October
Year: 2022
Address: Gyeongju, Republic of Korea
Venue: COLING
SIG: –
Publisher: International Committee on Computational Linguistics
Note: –
Pages: 4593–4603
Language: –
URL: https://aclanthology.org/2022.coling-1.406
DOI: –
Bibkey: `mosca-etal-2022-shap`
Cite (ACL): Edoardo Mosca, Ferenc Szigeti, Stella Tragianni, Daniel Gallagher, and Georg Groh. 2022. SHAP-Based Explanation Methods: A Review for NLP Interpretability. In *Proceedings of the 29th International Conference on Computational Linguistics*, pages 4593–4603, Gyeongju, Republic of Korea. International Committee on Computational Linguistics.
Cite (Informal): SHAP-Based Explanation Methods: A Review for NLP Interpretability (Mosca et al., COLING 2022)
Copy Citation: BibTeX Markdown MODS XML Endnote More options...
PDF: https://aclanthology.org/2022.coling-1.406.pdf

ACL materials are Copyright © 1963–2023 ACL; other materials are copyrighted by their respective copyright holders. Materials prior to 2016 here are licensed under the Creative Commons Attribution-NonCommercial-ShareAlike 3.0 International License. Permission is granted to make copies for the purposes of teaching and research. Materials published in or after 2016 are licensed on a Creative Commons Attribution 4.0 International License.

The ACL Anthology is managed and built by the ACL Anthology team of volunteers.

Site last built on 10 August 2023 at 17:55 UTC with commit 0c51aa7.

Figure 1: Screenshot for Study I on the ACL anthology, accessed on 22/11/2023 at 15:00, publicly available also at https://aclanthology.org/2022.coling-1.406/.

[1] https://aclanthology.org/

English Search **Donate** Explore CC

WHO WE ARE **WHAT WE DO** **LICENSES AND TOOLS** **BLOG** **SUPPORT US**

CC is a small nonprofit fighting for the open web. We need your support to continue our work. **DONATE TODAY!**

CC BY 4.0 DEED

Attribution 4.0 International

See the legal code

You are free to:

Share — copy and redistribute the material in any medium or

format for any purpose, even commercially.

Adapt — remix, transform, and build upon the material for any purpose, even commercially.

The licensor cannot revoke these freedoms as long as you follow the license terms.

Under the following terms:

Attribution - You must give appropriate credit , provide a link to the license, and indicate if changes were made . You may do so in any reasonable manner, but not in any way that suggests the licensor endorses you or your use.

No additional restrictions - You may not apply legal terms or technological measures that legally restrict others from

doing anything the license permits.

Notices:

You do not have to comply with the license for elements of the material in the public domain or where your use is permitted by an applicable exception or limitation .

No warranties are given. The license may not give you all of the permissions necessary for your intended use. For example, other rights such as publicity, privacy, or moral rights may limit how you use the material.

Notice

This deed highlights only some of the key features and terms of the actual license. It is not a license and has no legal value. You should carefully review all of the terms and conditions of the

actual license before using the licensed material.

Creative Commons is not a law firm and does not provide legal services. Distributing, displaying, or linking to this deed or the license that it summarizes does not create a lawyer-client or any other relationship.

Creative Commons is the nonprofit behind the open licenses and other legal tools that allow creators to share their work. Our legal tools are free to use.

- Learn more about our work
- **Learn more about CC Licensing**
- Support our work
- Use the license for your own material.
- Licenses List
- Public Domain List

Contact Newsletter Privacy Policies Terms

CONTACT US

Creative Commons PO Box 1866,
Mountain View, CA 94042

info@creativecommons.org

+1-415-429-6753

SUBSCRIBE TO OUR NEWSLETTER

Your e SUBSCRIBE

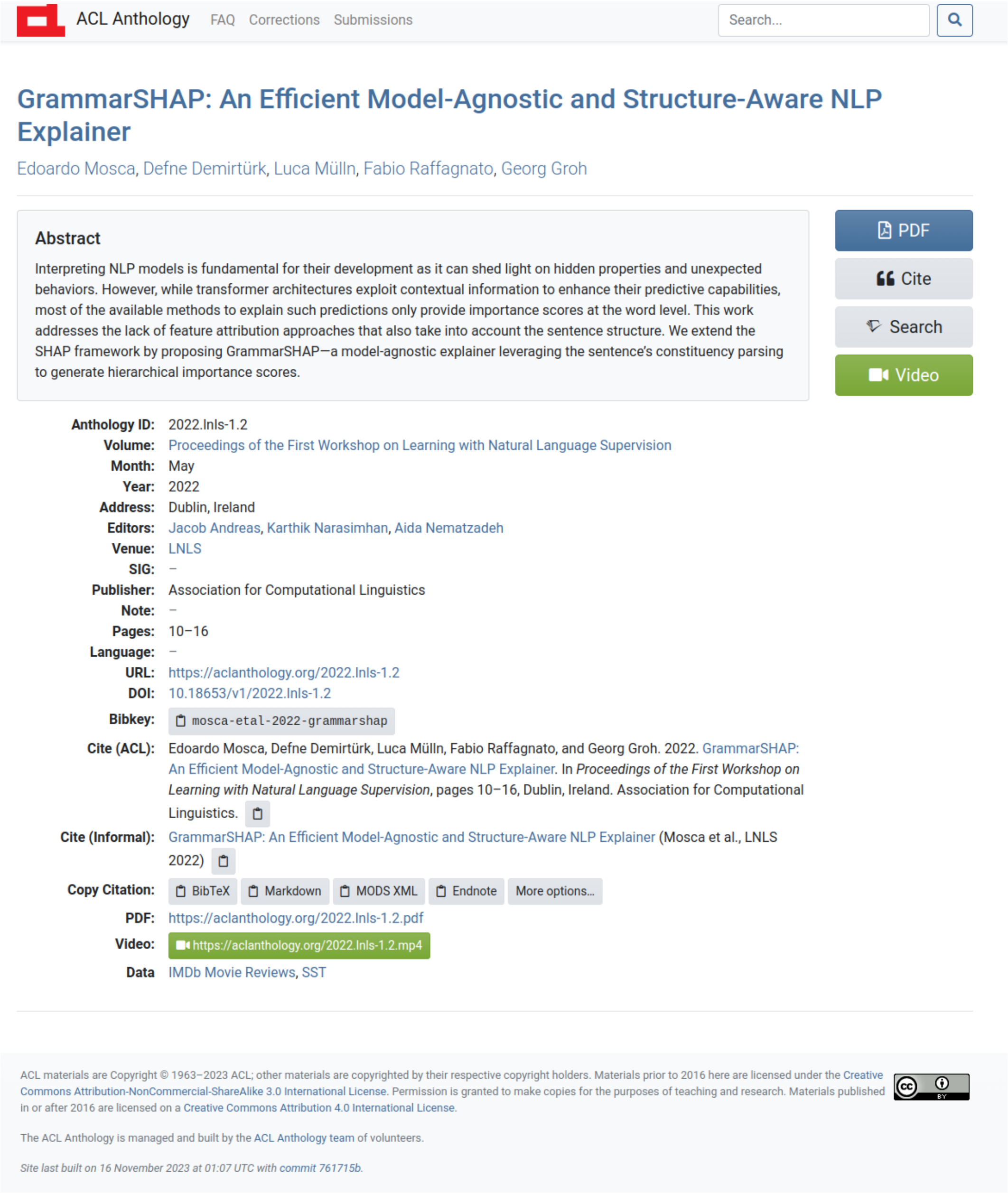
SUPPORT OUR WORK

Our work relies on you! Help us keep the Internet free and open.

DONATE NOW

2.2 Study II

Study II is published through a venue part of the *Association for Computational Linguistics* (ACL) and is public on the ACL Anthology website[2]. All articles in the anthology are published under the Creative Commons 4.0 Deed license (see Figure 2 at the bottom). Such license (CC BY 4.0 Deed) allows copying and redistributing the material in any medium or format for any purpose, even commercially. We attach the corresponding .pdf, retrieved from the website https://creativecommons.org/licenses/by/4.0/.



GrammarSHAP: An Efficient Model-Agnostic and Structure-Aware NLP Explainer

Edoardo Mosca, Defne Demirtürk, Luca Mülln, Fabio Raffagnato, Georg Groh

Abstract

Interpreting NLP models is fundamental for their development as it can shed light on hidden properties and unexpected behaviors. However, while transformer architectures exploit contextual information to enhance their predictive capabilities, most of the available methods to explain such predictions only provide importance scores at the word level. This work addresses the lack of feature attribution approaches that also take into account the sentence structure. We extend the SHAP framework by proposing GrammarSHAP—a model-agnostic explainer leveraging the sentence's constituency parsing to generate hierarchical importance scores.

Anthology ID:	2022.lnls-1.2
Volume:	Proceedings of the First Workshop on Learning with Natural Language Supervision
Month:	May
Year:	2022
Address:	Dublin, Ireland
Editors:	Jacob Andreas, Karthik Narasimhan, Aida Nematzadeh
Venue:	LNLS
SIG:	–
Publisher:	Association for Computational Linguistics
Note:	–
Pages:	10–16
Language:	–
URL:	https://aclanthology.org/2022.lnls-1.2
DOI:	10.18653/v1/2022.lnls-1.2
Bibkey:	mosca-etal-2022-grammarshap
Cite (ACL):	Edoardo Mosca, Defne Demirtürk, Luca Mülln, Fabio Raffagnato, and Georg Groh. 2022. GrammarSHAP: An Efficient Model-Agnostic and Structure-Aware NLP Explainer. In *Proceedings of the First Workshop on Learning with Natural Language Supervision*, pages 10–16, Dublin, Ireland. Association for Computational Linguistics.
Cite (Informal):	GrammarSHAP: An Efficient Model-Agnostic and Structure-Aware NLP Explainer (Mosca et al., LNLS 2022)
Copy Citation:	BibTeX Markdown MODS XML Endnote More options...
PDF:	https://aclanthology.org/2022.lnls-1.2.pdf
Video:	https://aclanthology.org/2022.lnls-1.2.mp4
Data	IMDb Movie Reviews, SST

The ACL Anthology is managed and built by the ACL Anthology team of volunteers.

Site last built on 16 November 2023 at 01:07 UTC with commit 761715b.

Figure 2: Screenshot for Study II on the ACL anthology, accessed on 22/11/2023 at 15:00, publicly available also at https://aclanthology.org/2022.lnls-1.2/.

[2] https://aclanthology.org/

English ⌄ Search Donate Explore CC

WHO WE ARE WHAT WE DO LICENSES AND TOOLS BLOG SUPPORT US

CC BY 4.0 DEED

Attribution 4.0 International

See the legal code

You are free to:

Share — copy and redistribute the material in any medium or

format for any purpose, even commercially.

Adapt — remix, transform, and build upon the material for any purpose, even commercially.

The licensor cannot revoke these freedoms as long as you follow the license terms.

Under the following terms:

Attribution - You must give appropriate credit , provide a link to the license, and indicate if changes were made . You may do so in any reasonable manner, but not in any way that suggests the licensor endorses you or your use.

No additional restrictions - You may not apply legal terms or technological measures that legally restrict others from

doing anything the license permits.

Notices:

You do not have to comply with the license for elements of the material in the public domain or where your use is permitted by an applicable exception or limitation .

No warranties are given. The license may not give you all of the permissions necessary for your intended use. For example, other rights such as publicity, privacy, or moral rights may limit how you use the material.

Notice

This deed highlights only some of the key features and terms of the actual license. It is not a license and has no legal value. You should carefully review all of the terms and conditions of the

actual license before using the licensed material.

Creative Commons is not a law firm and does not provide legal services. Distributing, displaying, or linking to this deed or the license that it summarizes does not create a lawyer-client or any other relationship.

Creative Commons is the nonprofit behind the open licenses and other legal tools that allow creators to share their work. Our legal tools are free to use.

- Learn more about our work
- **Learn more about CC Licensing**
- Support our work
- Use the license for your own material.
- Licenses List
- Public Domain List

Contact Newsletter Privacy Policies Terms

CONTACT US

Creative Commons PO Box 1866,
Mountain View, CA 94042

info@creativecommons.org

+1-415-429-6753

SUBSCRIBE TO OUR NEWSLETTER

Your e SUBSCRIBE

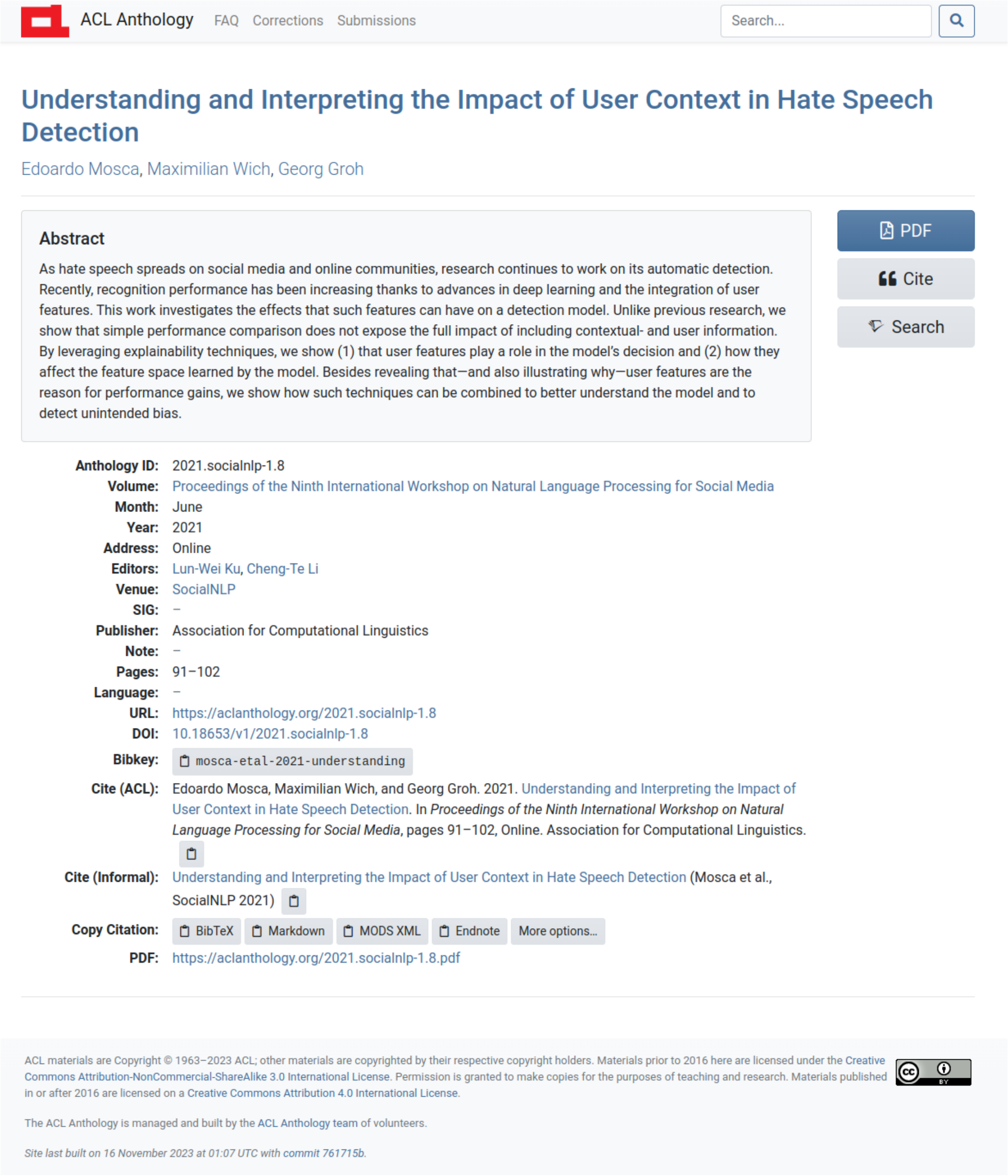
SUPPORT OUR WORK

Our work relies on you! Help us keep the Internet free and open.

DONATE NOW

2.3 Study III

Study III is published through a venue part of the *Association for Computational Linguistics* (ACL) and is public on the ACL Anthology website[3]. All articles in the anthology are published under the Creative Commons 4.0 Deed license (see Figure 3 at the bottom). Such license (CC BY 4.0 Deed) allows copying and redistributing the material in any medium or format for any purpose, even commercially. We attach the corresponding .pdf, retrieved from the website https://creativecommons.org/licenses/by/4.0/.



Understanding and Interpreting the Impact of User Context in Hate Speech Detection

Edoardo Mosca, Maximilian Wich, Georg Groh

Abstract

As hate speech spreads on social media and online communities, research continues to work on its automatic detection. Recently, recognition performance has been increasing thanks to advances in deep learning and the integration of user features. This work investigates the effects that such features can have on a detection model. Unlike previous research, we show that simple performance comparison does not expose the full impact of including contextual- and user information. By leveraging explainability techniques, we show (1) that user features play a role in the model's decision and (2) how they affect the feature space learned by the model. Besides revealing that—and also illustrating why—user features are the reason for performance gains, we show how such techniques can be combined to better understand the model and to detect unintended bias.

Anthology ID:	2021.socialnlp-1.8
Volume:	Proceedings of the Ninth International Workshop on Natural Language Processing for Social Media
Month:	June
Year:	2021
Address:	Online
Editors:	Lun-Wei Ku, Cheng-Te Li
Venue:	SocialNLP
SIG:	–
Publisher:	Association for Computational Linguistics
Note:	–
Pages:	91–102
Language:	–
URL:	https://aclanthology.org/2021.socialnlp-1.8
DOI:	10.18653/v1/2021.socialnlp-1.8
Bibkey:	`mosca-etal-2021-understanding`
Cite (ACL):	Edoardo Mosca, Maximilian Wich, and Georg Groh. 2021. Understanding and Interpreting the Impact of User Context in Hate Speech Detection. In *Proceedings of the Ninth International Workshop on Natural Language Processing for Social Media*, pages 91–102, Online. Association for Computational Linguistics.
Cite (Informal):	Understanding and Interpreting the Impact of User Context in Hate Speech Detection (Mosca et al., SocialNLP 2021)
Copy Citation:	BibTeX Markdown MODS XML Endnote More options…
PDF:	https://aclanthology.org/2021.socialnlp-1.8.pdf

The ACL Anthology is managed and built by the ACL Anthology team of volunteers.

Site last built on 16 November 2023 at 01:07 UTC with commit 761715b.

Figure 3: Screenshot for Study III on the ACL anthology, accessed on 22/11/2023 at 15:00, publicly available also at https://aclanthology.org/2021.socialnlp-1.8/.

[3] https://aclanthology.org/

English Search Donate Explore CC

WHO WE ARE WHAT WE DO LICENSES AND TOOLS BLOG SUPPORT US

CC BY 4.0 DEED

Attribution 4.0 International

See the legal code

You are free to:

Share — copy and redistribute the material in any medium or

format for any purpose, even commercially.

Adapt — remix, transform, and build upon the material for any purpose, even commercially.

The licensor cannot revoke these freedoms as long as you follow the license terms.

Under the following terms:

Attribution - You must give appropriate credit , provide a link to the license, and indicate if changes were made . You may do so in any reasonable manner, but not in any way that suggests the licensor endorses you or your use.

No additional restrictions - You may not apply legal terms or technological measures that legally restrict others from

doing anything the license permits.

Notices:

You do not have to comply with the license for elements of the material in the public domain or where your use is permitted by an applicable exception or limitation .

No warranties are given. The license may not give you all of the permissions necessary for your intended use. For example, other rights such as publicity, privacy, or moral rights may limit how you use the material.

Notice

This deed highlights only some of the key features and terms of the actual license. It is not a license and has no legal value. You should carefully review all of the terms and conditions of the

actual license before using the licensed material.

Creative Commons is not a law firm and does not provide legal services. Distributing, displaying, or linking to this deed or the license that it summarizes does not create a lawyer-client or any other relationship.

Creative Commons is the nonprofit behind the open licenses and other legal tools that allow creators to share their work. Our legal tools are free to use.

- Learn more about our work
- **Learn more about CC Licensing**
- Support our work
- Use the license for your own material.
- Licenses List
- Public Domain List

Contact Newsletter Privacy Policies Terms

CONTACT US

Creative Commons PO Box 1866,
Mountain View, CA 94042

info@creativecommons.org

+1-415-429-6753

SUBSCRIBE TO OUR NEWSLETTER

Your e SUBSCRIBE

SUPPORT OUR WORK

Our work relies on you! Help us keep the Internet free and open.

DONATE NOW

2.4 Study IV

Study IV is reprinted with permission from ©2021 Springer Nature Switzerland AG. We attach the .pdf confirmation in the following.

SPRINGER NATURE LICENSE
TERMS AND CONDITIONS

Aug 16, 2023

This Agreement between Mr. Edoardo Mosca ("You") and Springer Nature ("Springer Nature") consists of your license details and the terms and conditions provided by Springer Nature and Copyright Clearance Center.

License Number	5591951486069
License date	Jul 18, 2023
Licensed Content Publisher	Springer Nature
Licensed Content Publication	Springer eBook
Licensed Content Title	Explainable Abusive Language Classification Leveraging User and Network Data
Licensed Content Author	Maximilian Wich, Edoardo Mosca, Adrian Gorniak et al
Licensed Content Date	Jan 1, 2021
Type of Use	Thesis/Dissertation
Requestor type	academic/university or research institute
Format	electronic
Portion	full article/chapter
Will you be translating?	no
Circulation/distribution	1 - 29

Author of this Springer Nature content	yes
Title	Explainable Abusive Language Classification Leveraging User and Network Data
Institution name	Technical University of Munich
Expected presentation date	Oct 2023
Requestor Location	Mr. Edoardo Mosca Dachauer Straße 147 Munich, 80335 Germany Attn: Mr. Edoardo Mosca
Billing Type	Invoice
Billing Address	Mr. Edoardo Mosca Dachauer Straße 147 Munich, Germany 80335 Attn: Mr. Edoardo Mosca
Total	0.00 EUR
Terms and Conditions	

Springer Nature Customer Service Centre GmbH Terms and Conditions

The following terms and conditions ("Terms and Conditions") together with the terms specified in your [RightsLink] constitute the License ("License") between you as Licensee and Springer Nature Customer Service Centre GmbH as Licensor. By clicking 'accept' and completing the transaction for your use of the material ("Licensed Material"), you confirm your acceptance of and obligation to be bound by these Terms and Conditions.

1. Grant and Scope of License

1. 1. The Licensor grants you a personal, non-exclusive, non-transferable, non-sublicensable, revocable, world-wide License to reproduce, distribute, communicate to the public, make available, broadcast, electronically transmit or create derivative works using the Licensed Material for the purpose(s) specified in your RightsLink Licence Details only. Licenses are granted for the specific use requested in the order and for no other use, subject to these Terms and Conditions. You acknowledge and

agree that the rights granted to you under this License do not include the right to modify, edit, translate, include in collective works, or create derivative works of the Licensed Material in whole or in part unless expressly stated in your RightsLink Licence Details. You may use the Licensed Material only as permitted under this Agreement and will not reproduce, distribute, display, perform, or otherwise use or exploit any Licensed Material in any way, in whole or in part, except as expressly permitted by this License.

1. 2. You may only use the Licensed Content in the manner and to the extent permitted by these Terms and Conditions, by your RightsLink Licence Details and by any applicable laws.

1. 3. A separate license may be required for any additional use of the Licensed Material, e.g. where a license has been purchased for print use only, separate permission must be obtained for electronic re-use. Similarly, a License is only valid in the language selected and does not apply for editions in other languages unless additional translation rights have been granted separately in the License.

1. 4. Any content within the Licensed Material that is owned by third parties is expressly excluded from the License.

1. 5. Rights for additional reuses such as custom editions, computer/mobile applications, film or TV reuses and/or any other derivative rights requests require additional permission and may be subject to an additional fee. Please apply to journalpermissions@springernature.com or bookpermissions@springernature.com for these rights.

2. Reservation of Rights

Licensor reserves all rights not expressly granted to you under this License. You acknowledge and agree that nothing in this License limits or restricts Licensor's rights in or use of the Licensed Material in any way. Neither this License, nor any act, omission, or statement by Licensor or you, conveys any ownership right to you in any Licensed Material, or to any element or portion thereof. As between Licensor and you, Licensor owns and retains all right, title, and interest in and to the Licensed Material subject to the license granted in Section 1.1. Your permission to use the Licensed Material is expressly conditioned on you not impairing Licensor's or the applicable copyright owner's rights in the Licensed Material in any way.

3. Restrictions on use

3. 1. Minor editing privileges are allowed for adaptations for stylistic purposes or formatting purposes provided such alterations do not alter the original meaning or intention of the Licensed Material and the new figure(s) are still accurate and representative of the Licensed Material. Any other changes including but not limited to, cropping, adapting, and/or omitting material that affect the meaning, intention or moral rights of the author(s) are strictly prohibited.

3. 2. You must not use any Licensed Material as part of any design or trademark.

3. 3. Licensed Material may be used in Open Access Publications (OAP), but any such reuse must include a clear acknowledgment of this permission visible at the same time as the figures/tables/illustration or abstract and which must indicate that the Licensed Material is not part of the governing OA license but has been reproduced with permission. This may be indicated according to any standard referencing system but must include at a minimum 'Book/Journal title, Author, Journal Name (if applicable), Volume (if applicable), Publisher, Year, reproduced with permission from SNCSC'.

4. STM Permission Guidelines

4. 1. An alternative scope of license may apply to signatories of the STM Permissions Guidelines ("STM PG") as amended from time to time and made available at https://www.stm-assoc.org/intellectual-property/permissions/permissions-guidelines/.

4. 2. For content reuse requests that qualify for permission under the STM PG, and which may be updated from time to time, the STM PG supersede the terms and conditions contained in this License.

4. 3. If a License has been granted under the STM PG, but the STM PG no longer apply at the time of publication, further permission must be sought from the Rightsholder. Contact journalpermissions@springernature.com or bookpermissions@springernature.com for these rights.

5. Duration of License

5. 1. Unless otherwise indicated on your License, a License is valid from the date of purchase ("License Date") until the end of the relevant period in the below table:

Reuse in a medical communications project	Reuse up to distribution or time period indicated in License
Reuse in a dissertation/thesis	Lifetime of thesis
Reuse in a journal/magazine	Lifetime of journal/magazine
Reuse in a book/textbook	Lifetime of edition
Reuse on a website	1 year unless otherwise specified in the License
Reuse in a presentation/slide kit/poster	Lifetime of presentation/slide kit/poster. Note: publication whether electronic or in print of presentation/slide kit/poster may require further permission.
Reuse in conference proceedings	Lifetime of conference proceedings
Reuse in an annual report	Lifetime of annual report
Reuse in training/CME materials	Reuse up to distribution or time period indicated in License
Reuse in newsmedia	Lifetime of newsmedia
Reuse in coursepack/classroom materials	Reuse up to distribution and/or time period indicated in license

6. Acknowledgement

6. 1. The Licensor's permission must be acknowledged next to the Licensed Material in print. In electronic form, this acknowledgement must be visible at the same time as the figures/tables/illustrations or abstract and must be hyperlinked to the journal/book's homepage.

6. 2. Acknowledgement may be provided according to any standard referencing system and at a minimum should include "Author, Article/Book Title, Journal name/Book imprint, volume, page number, year, Springer Nature".

7. Reuse in a dissertation or thesis

7. 1. Where 'reuse in a dissertation/thesis' has been selected, the following terms apply: Print rights of the Version of Record are provided for; electronic rights for use only on institutional repository as defined by the Sherpa guideline (www.sherpa.ac.uk/romeo/) and only up to what is required by the awarding institution.

7. 2. For theses published under an ISBN or ISSN, separate permission is required. Please contact journalpermissions@springernature.com or bookpermissions@springernature.com for these rights.

7. 3. Authors must properly cite the published manuscript in their thesis according to current citation standards and include the following acknowledgement: '*Reproduced with permission from Springer Nature*'.

8. License Fee

You must pay the fee set forth in the License Agreement (the "License Fees"). All amounts payable by you under this License are exclusive of any sales, use, withholding, value added or similar taxes, government fees or levies or other assessments. Collection and/or remittance of such taxes to the relevant tax authority shall be the responsibility of the party who has the legal obligation to do so.

9. Warranty

9. 1. The Licensor warrants that it has, to the best of its knowledge, the rights to license reuse of the Licensed Material. **You are solely responsible for ensuring that the material you wish to license is original to the Licensor and does not carry the copyright of another entity or third party (as credited in the published version).** If the credit line on any part of the Licensed Material indicates that it was reprinted or adapted with permission from another source, then you should seek additional permission from that source to reuse the material.

9. 2. EXCEPT FOR THE EXPRESS WARRANTY STATED HEREIN AND TO THE EXTENT PERMITTED BY APPLICABLE LAW, LICENSOR PROVIDES THE LICENSED MATERIAL "AS IS" AND MAKES NO OTHER REPRESENTATION OR WARRANTY. LICENSOR EXPRESSLY DISCLAIMS ANY LIABILITY FOR ANY CLAIM ARISING FROM OR OUT OF THE CONTENT, INCLUDING BUT NOT LIMITED TO ANY ERRORS, INACCURACIES, OMISSIONS, OR DEFECTS CONTAINED THEREIN, AND ANY IMPLIED OR EXPRESS WARRANTY AS TO MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. IN NO EVENT SHALL LICENSOR BE LIABLE TO YOU OR ANY OTHER PARTY OR ANY OTHER PERSON OR FOR ANY SPECIAL, CONSEQUENTIAL, INCIDENTAL, INDIRECT, PUNITIVE, OR EXEMPLARY DAMAGES, HOWEVER CAUSED, ARISING OUT OF OR IN CONNECTION WITH THE DOWNLOADING, VIEWING OR USE OF THE LICENSED MATERIAL REGARDLESS OF THE FORM OF ACTION, WHETHER FOR BREACH OF CONTRACT, BREACH OF WARRANTY, TORT, NEGLIGENCE, INFRINGEMENT OR OTHERWISE (INCLUDING, WITHOUT LIMITATION, DAMAGES BASED ON LOSS OF PROFITS, DATA, FILES, USE, BUSINESS OPPORTUNITY OR CLAIMS OF THIRD PARTIES), AND WHETHER OR NOT THE PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES. THIS LIMITATION APPLIES NOTWITHSTANDING ANY FAILURE OF ESSENTIAL PURPOSE OF ANY LIMITED REMEDY PROVIDED HEREIN.

10. Termination and Cancellation

10. 1. The License and all rights granted hereunder will continue until the end of the applicable period shown in Clause 5.1 above. Thereafter, this license will be

terminated and all rights granted hereunder will cease.

10. 2. Licensor reserves the right to terminate the License in the event that payment is not received in full or if you breach the terms of this License.

11. General

11. 1. The License and the rights and obligations of the parties hereto shall be construed, interpreted and determined in accordance with the laws of the Federal Republic of Germany without reference to the stipulations of the CISG (United Nations Convention on Contracts for the International Sale of Goods) or to Germany´s choice-of-law principle.

11. 2. The parties acknowledge and agree that any controversies and disputes arising out of this License shall be decided exclusively by the courts of or having jurisdiction for Heidelberg, Germany, as far as legally permissible.

11. 3. This License is solely for Licensor's and Licensee's benefit. It is not for the benefit of any other person or entity.

Questions? For questions on Copyright Clearance Center accounts or website issues please contact springernaturesupport@copyright.com or +1-855-239-3415 (toll free in the US) or +1-978-646-2777. For questions on Springer Nature licensing please visit https://www.springernature.com/gp/partners/rights-permissions-third-party-distribution

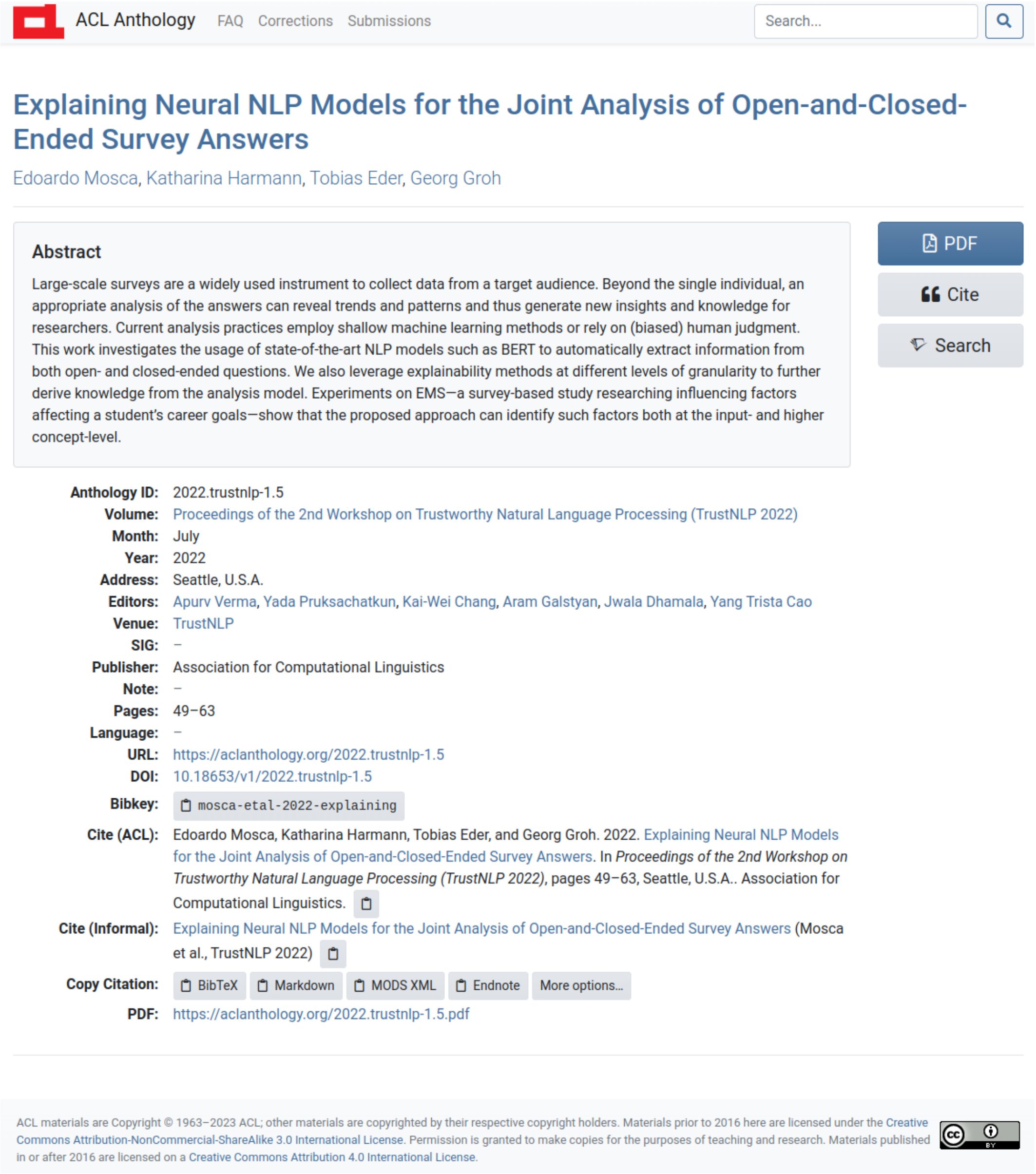
Other Conditions:

Version 1.4 - Dec 2022

Questions? customercare@copyright.com.

2.5 Study V

Study V is published through a venue part of the *Association for Computational Linguistics* (ACL) and is public on the ACL Anthology website[4]. All articles in the anthology are published under the Creative Commons 4.0 Deed license (see Figure 4 at the bottom). Such license (CC BY 4.0 Deed) allows copying and redistributing the material in any medium or format for any purpose, even commercially. We attach the corresponding .pdf, retrieved from the website https://creativecommons.org/licenses/by/4.0/.



Explaining Neural NLP Models for the Joint Analysis of Open-and-Closed-Ended Survey Answers

Edoardo Mosca, Katharina Harmann, Tobias Eder, Georg Groh

Abstract

Large-scale surveys are a widely used instrument to collect data from a target audience. Beyond the single individual, an appropriate analysis of the answers can reveal trends and patterns and thus generate new insights and knowledge for researchers. Current analysis practices employ shallow machine learning methods or rely on (biased) human judgment. This work investigates the usage of state-of-the-art NLP models such as BERT to automatically extract information from both open- and closed-ended questions. We also leverage explainability methods at different levels of granularity to further derive knowledge from the analysis model. Experiments on EMS—a survey-based study researching influencing factors affecting a student's career goals—show that the proposed approach can identify such factors both at the input- and higher concept-level.

Anthology ID:	2022.trustnlp-1.5
Volume:	Proceedings of the 2nd Workshop on Trustworthy Natural Language Processing (TrustNLP 2022)
Month:	July
Year:	2022
Address:	Seattle, U.S.A.
Editors:	Apurv Verma, Yada Pruksachatkun, Kai-Wei Chang, Aram Galstyan, Jwala Dhamala, Yang Trista Cao
Venue:	TrustNLP
SIG:	–
Publisher:	Association for Computational Linguistics
Note:	–
Pages:	49–63
Language:	–
URL:	https://aclanthology.org/2022.trustnlp-1.5
DOI:	10.18653/v1/2022.trustnlp-1.5
Bibkey:	mosca-etal-2022-explaining
Cite (ACL):	Edoardo Mosca, Katharina Harmann, Tobias Eder, and Georg Groh. 2022. Explaining Neural NLP Models for the Joint Analysis of Open-and-Closed-Ended Survey Answers. In *Proceedings of the 2nd Workshop on Trustworthy Natural Language Processing (TrustNLP 2022)*, pages 49–63, Seattle, U.S.A.. Association for Computational Linguistics.
Cite (Informal):	Explaining Neural NLP Models for the Joint Analysis of Open-and-Closed-Ended Survey Answers (Mosca et al., TrustNLP 2022)
Copy Citation:	BibTeX Markdown MODS XML Endnote More options...
PDF:	https://aclanthology.org/2022.trustnlp-1.5.pdf

Figure 4: Screenshot for Study V on the ACL anthology, accessed on 22/11/2023 at 15:00, publicly available also at https://aclanthology.org/2022.trustnlp-1.5/.

[4] https://aclanthology.org/

English ⌄ Search Donate Explore CC

WHO WE ARE WHAT WE DO LICENSES AND TOOLS BLOG SUPPORT US

CC BY 4.0 DEED

Attribution 4.0 International

See the legal code

You are free to:

Share — copy and redistribute the material in any medium or

format for any purpose, even commercially.

Adapt — remix, transform, and build upon the material for any purpose, even commercially.

The licensor cannot revoke these freedoms as long as you follow the license terms.

Under the following terms:

Attribution - You must give appropriate credit , provide a link to the license, and indicate if changes were made . You may do so in any reasonable manner, but not in any way that suggests the licensor endorses you or your use.

No additional restrictions - You may not apply legal terms or technological measures that legally restrict others from

doing anything the license permits.

Notices:

You do not have to comply with the license for elements of the material in the public domain or where your use is permitted by an applicable exception or limitation .

No warranties are given. The license may not give you all of the permissions necessary for your intended use. For example, other rights such as publicity, privacy, or moral rights may limit how you use the material.

Notice

This deed highlights only some of the key features and terms of the actual license. It is not a license and has no legal value. You should carefully review all of the terms and conditions of the

actual license before using the licensed material.

Creative Commons is not a law firm and does not provide legal services. Distributing, displaying, or linking to this deed or the license that it summarizes does not create a lawyer-client or any other relationship.

Creative Commons is the nonprofit behind the open licenses and other legal tools that allow creators to share their work. Our legal tools are free to use.

- Learn more about our work
- **Learn more about CC Licensing**
- Support our work
- Use the license for your own material.
- Licenses List
- Public Domain List

Contact Newsletter Privacy Policies Terms

CONTACT US

Creative Commons PO Box 1866,
Mountain View, CA 94042

info@creativecommons.org

+1-415-429-6753

SUBSCRIBE TO OUR NEWSLETTER

Your e

SUBSCRIBE

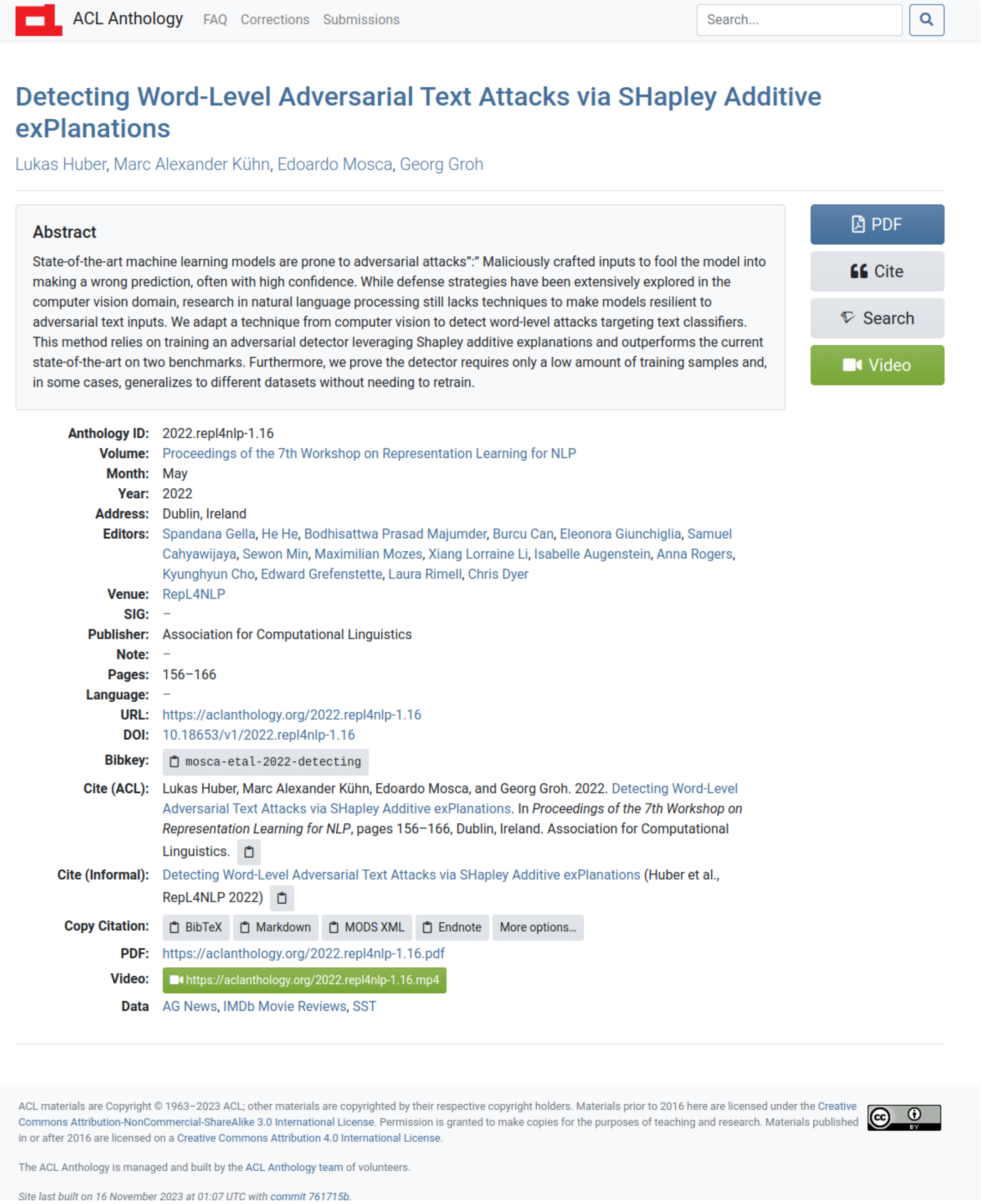
SUPPORT OUR WORK

Our work relies on you! Help us keep the Internet free and open.

DONATE NOW

2.6 Study VI

Study VI is published through a venue part of the *Association for Computational Linguistics* (ACL) and is public on the ACL Anthology website[5]. All articles in the anthology are published under the Creative Commons 4.0 Deed license (see Figure 5 at the bottom). Such license (CC BY 4.0 Deed) allows copying and redistributing the material in any medium or format for any purpose, even commercially. We attach the corresponding .pdf, retrieved from the website https://creativecommons.org/licenses/by/4.0/.



Detecting Word-Level Adversarial Text Attacks via SHapley Additive exPlanations

Lukas Huber, Marc Alexander Kühn, Edoardo Mosca, Georg Groh

Abstract

State-of-the-art machine learning models are prone to adversarial attacks”:” Maliciously crafted inputs to fool the model into making a wrong prediction, often with high confidence. While defense strategies have been extensively explored in the computer vision domain, research in natural language processing still lacks techniques to make models resilient to adversarial text inputs. We adapt a technique from computer vision to detect word-level attacks targeting text classifiers. This method relies on training an adversarial detector leveraging Shapley additive explanations and outperforms the current state-of-the-art on two benchmarks. Furthermore, we prove the detector requires only a low amount of training samples and, in some cases, generalizes to different datasets without needing to retrain.

Anthology ID:	2022.repl4nlp-1.16
Volume:	Proceedings of the 7th Workshop on Representation Learning for NLP
Month:	May
Year:	2022
Address:	Dublin, Ireland
Editors:	Spandana Gella, He He, Bodhisattwa Prasad Majumder, Burcu Can, Eleonora Giunchiglia, Samuel Cahyawijaya, Sewon Min, Maximilian Mozes, Xiang Lorraine Li, Isabelle Augenstein, Anna Rogers, Kyunghyun Cho, Edward Grefenstette, Laura Rimell, Chris Dyer
Venue:	RepL4NLP
SIG:	–
Publisher:	Association for Computational Linguistics
Note:	–
Pages:	156–166
Language:	–
URL:	https://aclanthology.org/2022.repl4nlp-1.16
DOI:	10.18653/v1/2022.repl4nlp-1.16
Bibkey:	mosca-etal-2022-detecting
Cite (ACL):	Lukas Huber, Marc Alexander Kühn, Edoardo Mosca, and Georg Groh. 2022. Detecting Word-Level Adversarial Text Attacks via SHapley Additive exPlanations. In *Proceedings of the 7th Workshop on Representation Learning for NLP*, pages 156–166, Dublin, Ireland. Association for Computational Linguistics.
Cite (Informal):	Detecting Word-Level Adversarial Text Attacks via SHapley Additive exPlanations (Huber et al., RepL4NLP 2022)
Copy Citation:	BibTeX Markdown MODS XML Endnote More options...
PDF:	https://aclanthology.org/2022.repl4nlp-1.16.pdf
Video:	https://aclanthology.org/2022.repl4nlp-1.16.mp4
Data	AG News, IMDb Movie Reviews, SST

ACL materials are Copyright © 1963–2023 ACL; other materials are copyrighted by their respective copyright holders. Materials prior to 2016 here are licensed under the Creative Commons Attribution-NonCommercial-ShareAlike 3.0 International License. Permission is granted to make copies for the purposes of teaching and research. Materials published in or after 2016 are licensed on a Creative Commons Attribution 4.0 International License.

The ACL Anthology is managed and built by the ACL Anthology team of volunteers.

Site last built on 16 November 2023 at 01:07 UTC with commit 761715b.

Figure 5: Screenshot for Study VI on the ACL anthology, accessed on 22/11/2023 at 15:00, publicly available also at https://aclanthology.org/2022.repl4nlp-1.16/.

[5] https://aclanthology.org/

English
Search Donate Explore CC

WHO WE ARE WHAT WE DO LICENSES AND TOOLS BLOG SUPPORT US

CC BY 4.0 DEED

Attribution 4.0 International

See the legal code

You are free to:

Share — copy and redistribute the material in any medium or

format for any purpose, even commercially.

Adapt — remix, transform, and build upon the material for any purpose, even commercially.

The licensor cannot revoke these freedoms as long as you follow the license terms.

Under the following terms:

Attribution - You must give appropriate credit , provide a link to the license, and indicate if changes were made . You may do so in any reasonable manner, but not in any way that suggests the licensor endorses you or your use.

No additional restrictions - You may not apply legal terms or technological measures that legally restrict others from

actual license before using the licensed material.

Creative Commons is not a law firm and does not provide legal services. Distributing, displaying, or linking to this deed or the license that it summarizes does not create a lawyer-client or any other relationship.

Creative Commons is the nonprofit behind the open licenses and other legal tools that allow creators to share their work. Our legal tools are free to use.

- Learn more about our work
- **Learn more about CC Licensing**
- Support our work
- Use the license for your own material.
- Licenses List
- Public Domain List

Contact Newsletter Privacy Policies Terms

CONTACT US

Creative Commons PO Box 1866,
Mountain View, CA 94042

info@creativecommons.org

+1-415-429-6753

SUBSCRIBE TO OUR NEWSLETTER

Your e SUBSCRIBE

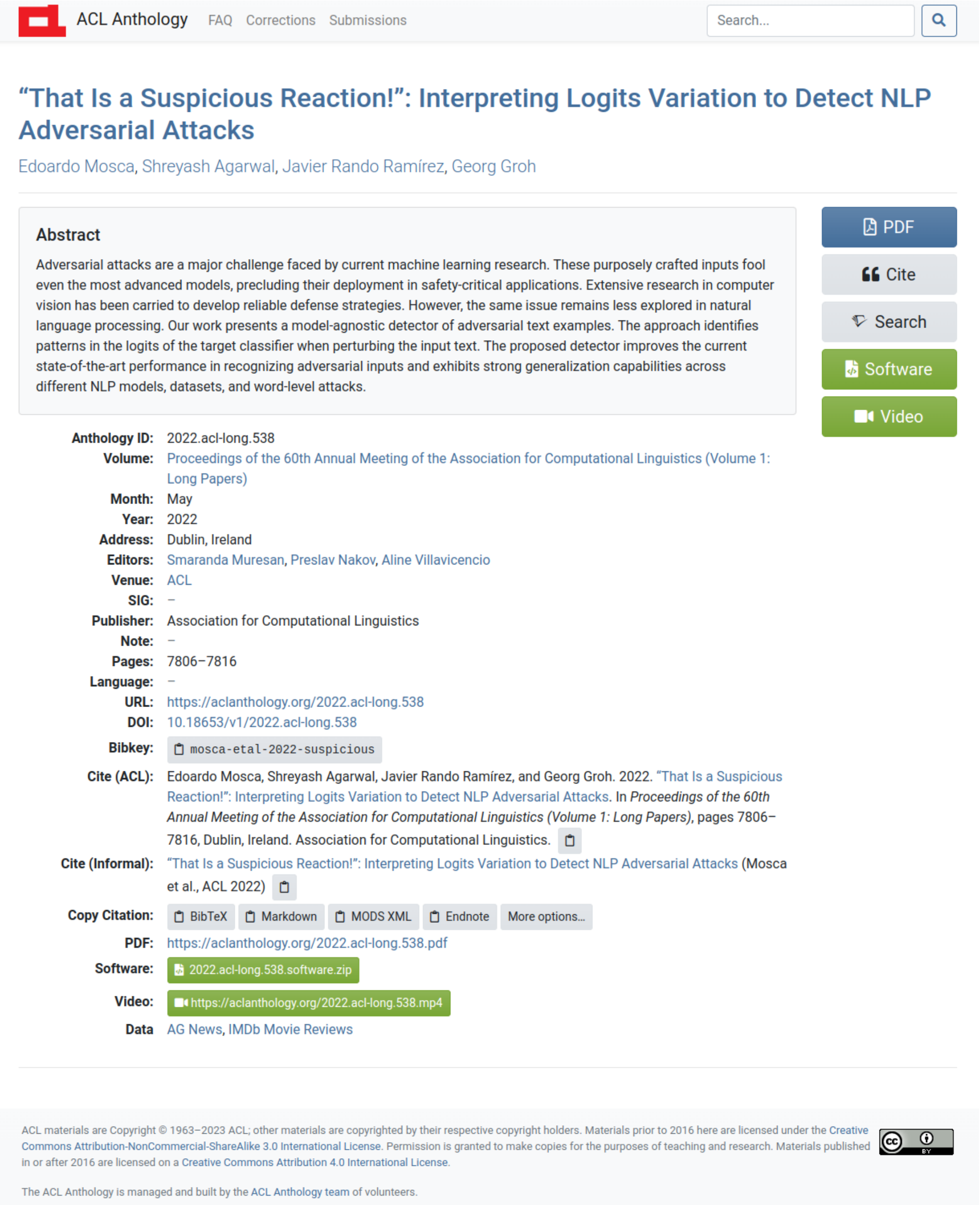
SUPPORT OUR WORK

Our work relies on you! Help us keep the Internet free and open.

DONATE NOW

2.7 Study VII

Study VII is published through a venue part of the *Association for Computational Linguistics* (ACL) and is public on the ACL Anthology website[6]. All articles in the anthology are published under the Creative Commons 4.0 Deed license (see Figure 6 at the bottom). Such license (CC BY 4.0 Deed) allows copying and redistributing the material in any medium or format for any purpose, even commercially. We attach the corresponding .pdf, retrieved from the website https://creativecommons.org/licenses/by/4.0/.



“That Is a Suspicious Reaction!”: Interpreting Logits Variation to Detect NLP Adversarial Attacks

Edoardo Mosca, Shreyash Agarwal, Javier Rando Ramírez, Georg Groh

Abstract

Adversarial attacks are a major challenge faced by current machine learning research. These purposely crafted inputs fool even the most advanced models, precluding their deployment in safety-critical applications. Extensive research in computer vision has been carried to develop reliable defense strategies. However, the same issue remains less explored in natural language processing. Our work presents a model-agnostic detector of adversarial text examples. The approach identifies patterns in the logits of the target classifier when perturbing the input text. The proposed detector improves the current state-of-the-art performance in recognizing adversarial inputs and exhibits strong generalization capabilities across different NLP models, datasets, and word-level attacks.

Anthology ID: 2022.acl-long.538
Volume: Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)
Month: May
Year: 2022
Address: Dublin, Ireland
Editors: Smaranda Muresan, Preslav Nakov, Aline Villavicencio
Venue: ACL
SIG: –
Publisher: Association for Computational Linguistics
Note: –
Pages: 7806–7816
Language: –
URL: https://aclanthology.org/2022.acl-long.538
DOI: 10.18653/v1/2022.acl-long.538
Bibkey: mosca-etal-2022-suspicious
Cite (ACL): Edoardo Mosca, Shreyash Agarwal, Javier Rando Ramírez, and Georg Groh. 2022. “That Is a Suspicious Reaction!”: Interpreting Logits Variation to Detect NLP Adversarial Attacks. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 7806–7816, Dublin, Ireland. Association for Computational Linguistics.
Cite (Informal): “That Is a Suspicious Reaction!”: Interpreting Logits Variation to Detect NLP Adversarial Attacks (Mosca et al., ACL 2022)
Copy Citation: BibTeX Markdown MODS XML Endnote More options...
PDF: https://aclanthology.org/2022.acl-long.538.pdf
Software: 2022.acl-long.538.software.zip
Video: https://aclanthology.org/2022.acl-long.538.mp4
Data AG News, IMDb Movie Reviews

The ACL Anthology is managed and built by the ACL Anthology team of volunteers.

Figure 6: Screenshot for Study VII on the ACL anthology, accessed on 22/11/2023 at 15:00, publicly available also at https://aclanthology.org/2022.acl-long.538/.

[6] https://aclanthology.org/

English Search Donate Explore CC

WHO WE ARE WHAT WE DO LICENSES AND TOOLS BLOG SUPPORT US

CC BY 4.0 DEED

Attribution 4.0 International

See the legal code

You are free to:

Share — copy and redistribute the material in any medium or

format for any purpose, even commercially.

Adapt — remix, transform, and build upon the material for any purpose, even commercially.

The licensor cannot revoke these freedoms as long as you follow the license terms.

Under the following terms:

Attribution - You must give appropriate credit , provide a link to the license, and indicate if changes were made . You may do so in any reasonable manner, but not in any way that suggests the licensor endorses you or your use.

No additional restrictions - You may not apply legal terms or technological measures that legally restrict others from

doing anything the license permits.

Notices:

You do not have to comply with the license for elements of the material in the public domain or where your use is permitted by an applicable exception or limitation .

No warranties are given. The license may not give you all of the permissions necessary for your intended use. For example, other rights such as publicity, privacy, or moral rights may limit how you use the material.

Notice

This deed highlights only some of the key features and terms of the actual license. It is not a license and has no legal value. You should carefully review all of the terms and conditions of the

actual license before using the licensed material.

Creative Commons is not a law firm and does not provide legal services. Distributing, displaying, or linking to this deed or the license that it summarizes does not create a lawyer-client or any other relationship.

Creative Commons is the nonprofit behind the open licenses and other legal tools that allow creators to share their work. Our legal tools are free to use.

- Learn more about our work
- **Learn more about CC Licensing**
- Support our work
- Use the license for your own material.
- Licenses List
- Public Domain List

Contact Newsletter Privacy Policies Terms

CONTACT US

Creative Commons PO Box 1866,
Mountain View, CA 94042

info@creativecommons.org

+1-415-429-6753

SUBSCRIBE TO OUR NEWSLETTER

Your e

SUBSCRIBE

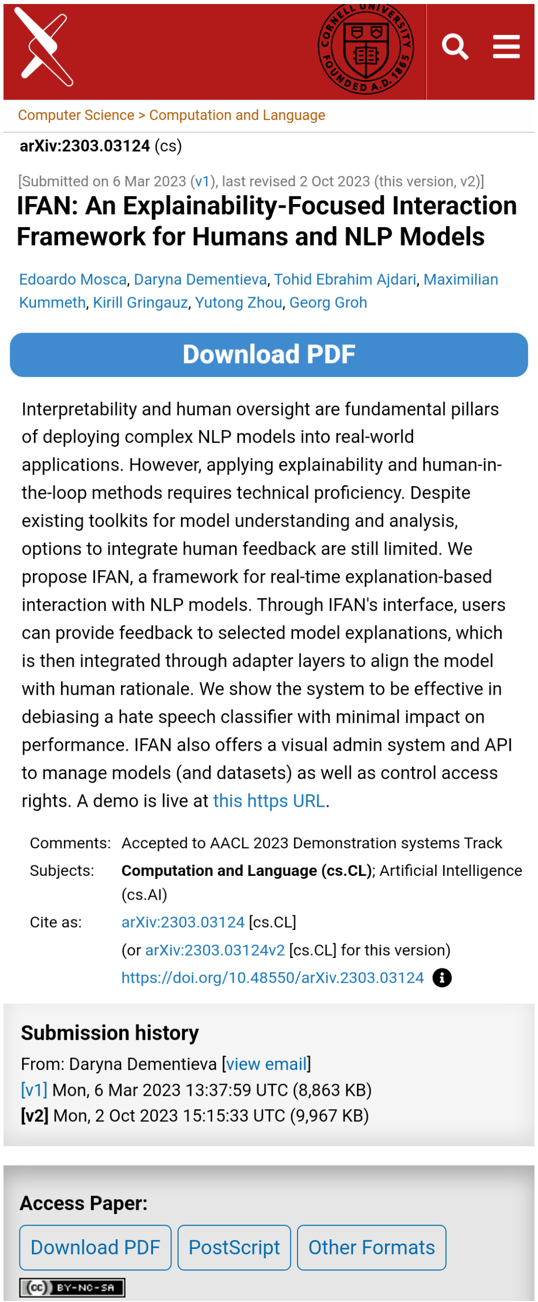
SUPPORT OUR WORK

Our work relies on you! Help us keep the Internet free and open.

DONATE NOW

2.8 Study VIII

Study VIII is publicly available at https://arxiv.org/abs/2303.03124, and published under the license Attribution-NonCommercial-ShareAlike 4.0 International (please notice the licensing button at the bottom left of Figure 6). Such license (CC BY-NC-SA 4.0) allows copying and redistributing the material in any medium or format for non-commercial purposes. We also attach the corresponding .pdf, retrieved from https://creativecommons.org/licenses/by-nc-sa/4.0/.



Computer Science > Computation and Language

arXiv:2303.03124 (cs)

[Submitted on 6 Mar 2023 (v1), last revised 2 Oct 2023 (this version, v2)]

IFAN: An Explainability-Focused Interaction Framework for Humans and NLP Models

Edoardo Mosca, Daryna Dementieva, Tohid Ebrahim Ajdari, Maximilian Kummeth, Kirill Gringauz, Yutong Zhou, Georg Groh

Interpretability and human oversight are fundamental pillars of deploying complex NLP models into real-world applications. However, applying explainability and human-in-the-loop methods requires technical proficiency. Despite existing toolkits for model understanding and analysis, options to integrate human feedback are still limited. We propose IFAN, a framework for real-time explanation-based interaction with NLP models. Through IFAN's interface, users can provide feedback to selected model explanations, which is then integrated through adapter layers to align the model with human rationale. We show the system to be effective in debiasing a hate speech classifier with minimal impact on performance. IFAN also offers a visual admin system and API to manage models (and datasets) as well as control access rights. A demo is live at this https URL.

Comments:	Accepted to AACL 2023 Demonstration systems Track
Subjects:	**Computation and Language (cs.CL)**; Artificial Intelligence (cs.AI)
Cite as:	arXiv:2303.03124 [cs.CL] (or arXiv:2303.03124v2 [cs.CL] for this version) https://doi.org/10.48550/arXiv.2303.03124

Submission history

From: Daryna Dementieva [view email]
[v1] Mon, 6 Mar 2023 13:37:59 UTC (8,863 KB)
[v2] Mon, 2 Oct 2023 15:15:33 UTC (9,967 KB)

Access Paper:

Figure 7: Screenshot for Study VIII on the arXiv, accessed on 22/11/2023 at 15:00, publicly available also at https://arxiv.org/abs/2303.03124.

English Search Donate Explore CC

WHO WE ARE WHAT WE DO LICENSES AND TOOLS BLOG SUPPORT US

CC is a small nonprofit fighting for the open web. We need your support to continue our work. DONATE TODAY!

CC BY-NC-SA 4.0 DEED

Attribution-NonCommercial-ShareAlike 4.0 International

See the legal code

You are free to:

Share — copy and redistribute the material in any medium or format

Adapt — remix, transform, and build upon the material

The licensor cannot revoke these freedoms as long as you follow the license terms.

Under the following terms:

Attribution - You must give appropriate credit , provide a link to the license, and indicate if changes were made . You may do so in any reasonable manner, but not in any way that suggests the licensor endorses you or your use.

NonCommercial - You may not use the material for commercial purposes .

Notices:

privacy, or moral rights may limit how you use the material.

Notice

This deed highlights only some of the key features and terms of the actual license. It is not a license and has no legal value. You should carefully review all of the terms and conditions of the actual license before using the licensed material.

Creative Commons is not a law firm and does not provide legal services. Distributing, displaying, or linking to this deed or the license that it summarizes does not create a lawyer-client or any other relationship.

Creative Commons is the nonprofit behind the open licenses and other legal tools that allow creators to share their work. Our legal tools are free to use.

- Learn more about our work

- **Learn more about CC Licensing**
- Support our work
- Use the license for your own material.
- Licenses List
- Public Domain List

Contact Newsletter Privacy Policies Terms

CONTACT US

Creative Commons PO Box 1866,
Mountain View, CA 94042

info@creativecommons.org

+1-415-429-6753

SUBSCRIBE TO OUR NEWSLETTER

Your e

SUBSCRIBE

SUPPORT OUR WORK

Our work relies on you! Help us keep the Internet free and open.

DONATE NOW

References

Huber, Lukas et al. (May 2022). "Detecting Word-Level Adversarial Text Attacks via SHapley Additive exPlanations". In: *Proceedings of the 7th Workshop on Representation Learning for NLP*. Dublin, Ireland: Association for Computational Linguistics, pp. 156–166. DOI: 10.18653/v1/2022.repl4nlp-1.16. URL: https://aclanthology.org/2022.repl4nlp-1.16.

Mosca, Edoardo, Shreyash Agarwal, et al. (May 2022). ""That Is a Suspicious Reaction!": Interpreting Logits Variation to Detect NLP Adversarial Attacks". In: *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Dublin, Ireland: Association for Computational Linguistics, pp. 7806–7816. DOI: 10.18653/v1/2022.acl-long.538. URL: https://aclanthology.org/2022.acl-long.538.

Mosca, Edoardo, Daryna Dementieva, et al. (2023). "IFAN: An Explainability-Focused Interaction Framework for Humans and NLP Models". In: *arXiv preprint arXiv:2303.03124*.

Mosca, Edoardo, Defne Demirtürk, et al. (May 2022). "GrammarSHAP: An Efficient Model-Agnostic and Structure-Aware NLP Explainer". In: *Proceedings of the First Workshop on Learning with Natural Language Supervision*. Dublin, Ireland: Association for Computational Linguistics, pp. 10–16. DOI: 10.18653/v1/2022.lnls-1.2. URL: https://aclanthology.org/2022.lnls-1.2.

Mosca, Edoardo, Katharina Harmann, et al. (July 2022). "Explaining Neural NLP Models for the Joint Analysis of Open-and-Closed-Ended Survey Answers". In: *Proceedings of the 2nd Workshop on Trustworthy Natural Language Processing (TrustNLP 2022)*. Seattle, U.S.A.: Association for Computational Linguistics, pp. 49–63. DOI: 10.18653/v1/2022.trustnlp-1.5. URL: https://aclanthology.org/2022.trustnlp-1.5.

Mosca, Edoardo, Ferenc Szigeti, et al. (Oct. 2022). "SHAP-Based Explanation Methods: A Review for NLP Interpretability". In: *Proceedings of the 29th International Conference on Computational Linguistics*. Gyeongju, Republic of Korea: International Committee on Computational Linguistics, pp. 4593–4603. URL: https://aclanthology.org/2022.coling-1.406.

Mosca, Edoardo, Maximilian Wich, and Georg Groh (June 2021). "Understanding and Interpreting the Impact of User Context in Hate Speech Detection". In: *Proceedings of the Ninth International Workshop on Natural Language Processing for Social Media*. Online: Association for Computational Linguistics, pp. 91–102. DOI: 10.18653/v1/2021.socialnlp-1.8. URL: https://aclanthology.org/2021.socialnlp-1.8.

Wich, Maximilian et al. (2021). "Explainable abusive language classification leveraging user and network data". In: *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*. Springer, pp. 481–496. URL: https://2021.ecmlpkdd.org/wp-content/uploads/2021/07/sub_663.pdf.