ferret: a Framework for Benchmarking Explainers on Transformers

Giuseppe Attanasio[♣], Eliana Pastor[♢], Chiara Di Bonaventura[♠], Debora Nozza[♣]

[♣]Bocconi University, Milan, Italy
[♢]Politecnico di Torino, Turin, Italy
[♠]King's College London, London, United Kingdom
{giuseppe.attanasio3,debora.nozza}@unibocconi.it
eliana.pastor@polito.it
chiara.di_bonaventura@kcl.ac.uk

Abstract

As Transformers are increasingly relied upon to solve complex NLP problems, there is an increased need for their decisions to be humanly interpretable. While several explainable AI (XAI) techniques for interpreting the outputs of transformer-based models have been proposed, there is still a lack of easy access to using and comparing them. We introduce *ferret*, a Python library to simplify the use and comparisons of XAI methods on transformer-based classifiers. With *ferret*, users can visualize and compare transformers-based models output explanations using state-of-the-art XAI methods on any free-text or existing XAI corpora. Moreover, users can also evaluate ad-hoc XAI metrics to select the most faithful and plausible explanations. To align with the recently consolidated process of sharing and using transformers-based models from Hugging Face, *ferret* interfaces directly with its Python library. In this paper, we showcase *ferret* to benchmark XAI methods used on transformers for sentiment analysis and hate speech detection. We show how specific methods provide consistently better explanations and are preferable in the context of transformer models.

1 Introduction

Transformers have revolutionized NLP applications in recent years due to their strong performance on various tasks; their black-box nature remains an obstacle for practitioners who need explanations about why specific predictions were made and what features drove them. The development of explainable AI (XAI) techniques on several NLP tasks (Madsen et al., 2022) has helped bridge this gap by providing insight into the inner workings of transformers and helping users gain trust in their decisions. Several XAI approaches have been proposed in the literature (Ribeiro et al., 2016; Lundberg and Lee, 2017; Simonyan et al., 2014a; Pastor and Baralis, 2019), also tailored to Transformer models (Wallace et al., 2019a; Li et al., 2016; Jin et al., 2019; Ross et al., 2021). Despite the importance of making XAI methods accessible to NLP experts and practitioners through practical tools, there is still a lack of accessibility for transformer models. XAI for transformers is mainly scattered and hard to operationalize. Methods come with independent implementations or framework-specific libraries that do not allow either evaluation or cross-method comparison. Further, existing implementations are not integrated with widespread transformers libraries (e.g., Hugging Face's *transformers* (Wolf et al., 2020)). The lack of standardization and weak interoperability leaves practitioners with unsolved questions, such as choosing the *best* method given a task and a model (Attanasio et al., 2022).

We introduce *ferret* (FramEwork foR benchmaRking Explainers on Transformers), an open-source Python library that drastically simplifies the use and comparison of XAI methods on transformers. The library stems from vertical scientific contributions and focused engineering efforts. On the one hand, *ferret* provides the first-of-its-kind API (see Figure 1) to use and compare explanation methods along the established criteria of faithfulness and plausibility (Jacovi and Goldberg, 2020). On the other hand, it integrates seamlessly with *transformers* (Wolf et al., 2020), making it an easy add-on to existing Transformer-based pipelines and NLP tasks. *ferret* permits to run four state-of-the-art XAI methods, compute six ad-hoc XAI evaluation metrics, and easily load four existing interpretability datasets. Further, it offers abstract interfaces to foster future integration of methods, metrics, and datasets.

We showcase *ferret* on sentiment analysis and

Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics
System Demonstrations, pages 256–266

hate speech detection case studies. Faithfulness and plausibility metrics highlight SHAP (Lundberg and Lee, 2017) as the most consistent explainer on single- and multiple-samples scenarios.

Contributions. We release *ferret*, the first-of-its-kind benchmarking framework for interpretability tightly integrated with Hugging Face's *transformers* library. We release our code and documentation,[1] an interactive demo,[2] and a video tutorial.[3]

2 Library Design

ferret builds on four core principles.

1. Built-in Post-hoc Interpretability We include four state-of-the-art post-hoc feature importance methods and three interpretability corpora. Ready-to-use methods allow users to explain any text with an arbitrary model. Annotated datasets provide valuable test cases for new interpretability methods and metrics. To the best of our knowledge, *ferret* is first in providing integrated access to XAI datasets, methods, and a full-fledged evaluation suite.

2. Unified Explanation Benchmarking We propose a unified API to evaluate explanations. We currently support six state-of-the metrics along the principles of faithfulness and plausibility (Jacovi and Goldberg, 2020).

3. Transformers-readiness *ferret* offers a direct interface with models from the Hugging Face Hub. Users can load models using standard naming conventions and explain them with the built-in methods effortlessly. Figure 1 shows the essential code to classify and explain a string with a pre-existing Hugging Face model and evaluate the resulting explanations.

4. Modularity and Abstraction *ferret* counts three core modules, implementing Explainers, Evaluation, and Datasets APIs. Each module exposes an abstract interface to foster new development. For example, user can sub-class `BaseExplainer` or `BaseEvaluator` to include a new feature importance method or a new evaluation metric respectively.

[1] https://github.com/g8a9/ferret
[2] https://huggingface.co/spaces/g8a9/ferret
[3] https://youtu.be/kX0HcSah_M4

Feature	Category
Gradient	Saliency
Integrated Gradient	Saliency
LIME	Surrogate Model
SHAP	Shapley Values
Comprehensiveness	Faithfulness
Sufficiency	Faithfulness
Correlation with Leave-One-Out scores	Faithfulness
Intersection-Over-Union	Plausibility
Area Under Precision-Recall Curve	Plausibility
Token-level F1 score	Plausibility
HateXplain	Hate Speech
MovieReviews	Sentiment
SST	Sentiment
Thermostat	Generic

Table 1: *ferret* at a glance: built-in methods (top), metrics (middle), and datasets (bottom).

ferret builds on common choices from the interpretability community and good engineering practices. We report the most salient technical details (e.g., efficiency via GPU inference, visualization tools, etc.) in Appendix A.

2.1 Explainer API

We focus on the widely adopted family of post-hoc feature attribution methods (Danilevsky et al., 2020). I.e., given a model, a target class, and a prediction, *ferret* lets you measure *how much* each token contributed to that prediction. We integrate Gradient (Simonyan et al., 2014b) (also known as Saliency) and Integrated Gradient (Sundararajan et al., 2017); SHAP (Lundberg and Lee, 2017) as a Shapley value-based method, and LIME (Ribeiro et al., 2016) as representative of local surrogate methods.

We build on open-source libraries and streamline their interaction with Hugging Face models and paradigms. We report the supported configurations and functionalities in Appendix A.

2.2 Dataset API

Fostering a streamlined, accessible evaluation on independently released XAI datasets, we provide a convenient Dataset API. It enables users to load XAI datasets, explain individual or subsets of samples, and evaluate the resulting explanations.

```python
from transformers import AutoModelForSequenceClassification, AutoTokenizer
from ferret import Benchmark

name = "cardiffnlp/twitter-xlm-roberta-base-sentiment"
model = AutoModelForSequenceClassification.from_pretrained(name)
tokenizer = AutoTokenizer.from_pretrained(name)

bench = Benchmark(model, tokenizer)
explanations = bench.explain("You look stunning!", target=1)
evaluations = bench.evaluate_explanations(explanations, target=1)
```

Figure 1: Essential code to benchmark explanations on an existing Hugging Face model using *ferret*.

Currently, *ferret* includes three classification-oriented datasets annotated with human rationales, i.e., annotations highlighting the most relevant words, phrases, or sentences a human annotator attributed to a given class label (DeYoung et al., 2020; Wiegreffe and Marasovic, 2021). Moreover, *ferret* API gives access to the Thermostat collection (Feldhus et al., 2021), a wide set of pre-computed feature attribution scores.

HateXplain (Mathew et al., 2021). It contains 20,148 English instances labeled along three axes: (i) hate (either hateful, offensive, normal or undecided), (ii) target group (either race, religion, gender, sexual orientation, or miscellaneous), and (iii) word-level human rationales (expressed only on hateful and offensive texts).[4]

MovieReviews (Zaidan and Eisner, 2008; DeYoung et al., 2020). The dataset contains 2,000 movie reviews annotated with positive and negative sentiment labels and phrase-level human rationales that support gold labels.

Stanford Sentiment Treebank (SST) (Socher et al., 2013). A sentiment classification dataset of 9,620 movie reviews annotated with binary sentiment labels, including human annotations for word phrases of the parse trees. We extract human rationales from annotations following the heuristic approach proposed in Carton et al. (2020).

Thermostat Datasets *Thermostat* (Feldhus et al., 2021) provides pre-computed feature attribution scores given a model, a dataset, and an explanation method. *ferret* currently provides built-in access to pre-computed attributions on the news topic classification and sentiment analysis tasks.

[4]If a model splits a relevant word into sub-words, we consider all of them relevant as well.

These datasets provide an *initial* example of what an integrated approach can offer to researchers and practitioners.

2.3 Evaluation API

We evaluate explanations on the faithfulness and plausibility properties (Jacovi and Goldberg, 2020; DeYoung et al., 2020). Specifically, *ferret* implements three state-of-the-art metrics to measure faithfulness and three for plausibility.

Faithfulness. Faithfulness evaluates how accurately the explanation reflects the inner working of the model (Jacovi and Goldberg, 2020).

ferret offers the following measures of faithfulness: comprehensiveness, sufficiency, (DeYoung et al., 2020) and correlations with 'leave-one-out' scores (Jain and Wallace, 2019).

Comprehensiveness (↑) evaluates whether the explanation captures the tokens the model used to make the prediction. We measure it by removing the tokens highlighted by the explainer and observing the change in probability as follows.

Let x be a sentence and let f_j be the prediction probability of the model f for a target class j. Let r_j be a discrete explanation or *rationale* indicating the set of tokens supporting the prediction f_j. Comprehensiveness is defined as $f(x)_j - f(x \setminus r_j)_j$ where $x \setminus r_j$ is the sentence x were tokens in r_j are removed. A high value of comprehensiveness indicates that the tokens in r_j are relevant for the prediction.

While comprehensiveness is defined for discrete explanations, feature attribution methods assign a continuous score to each token. We hence select identify r_j as follows. First, we filter out tokens with a negative contribution (i.e., they *pull* the prediction away from the chosen label). Then, we compute the metric multiple times, considering the $k\%$ most important tokens, with k ranging from

10% to 100% (step of 10%). Finally, we aggregate the comprehensiveness scores with the average, called Area Over the Perturbation Curve (AOPC) (DeYoung et al., 2020).

Sufficiency (↓) captures if the tokens in the explanation are sufficient for the model to make the prediction (DeYoung et al., 2020). It is measured as $f(x)_j - f(r_j)_j$. A low score indicates that tokens in r_j are indeed the ones driving the prediction. As for *Comprehensiveness*, we compute the AOPC by varying the number of the relevant tokens r_j.

Correlation with Leave-One-Out scores (↑). We first compute leave-one-out (LOO) scores by omitting tokens and measuring the difference in the model prediction. We do that for every token, once at a time. LOO scores represent a simple measure of individual feature importance under the linearity assumption (Jacovi and Goldberg, 2020). We then measure the Kendall rank correlation coefficient τ between the explanation and LOO importance (Jain and Wallace, 2019) ($taucorr\_loo$). $taucorr\_loo$ closer to 1 means higher faithfulness to LOO.

Plausibility. Plausibility reflects how explanations are aligned with human reasoning by comparing explanations with *human rationales* (DeYoung et al., 2020) .

We integrate into *ferret* three plausibility measures of the ERASER benchmark (DeYoung et al., 2020): Intersection-Over-Union (IOU) at the token level, token-level F1 scores, and Area Under the Precision-Recall curve (AUPRC).

The first two are defined for discrete explanations. Given the human and predicted rationale, *IOU* (↑) quantifies the overlap of the tokens they cover divided by the size of their union. *Token-level F1 scores* (↑) are derived by computing precision and recall at the token level. Following DeYoung et al. (2020) and Mathew et al. (2021), we derive discrete explanations by selecting the top K tokens with positive influence, where K is the average length of the human rationale for the dataset. While being intuitive, IOU and Token-level F1 are based only on a single threshold to derive rationales. Moreover, they do not consider tokens' relative ranking and degree of importance. We then also integrate the AUPRC (↑), defined for explanations with continuous scores (DeYoung et al., 2020). It is computed by varying a threshold over token importance scores, using the human rationale as ground truth.

2.4 Transformers-Ready Interface

ferret is deeply integrated with Hugging Face interfaces. Users working with their standard models and tokenizers can easily integrate it for diagnostic purposes. The contact point is the main `Benchmark` class. It receives any Hugging Face model and tokenizer and uses them to classify, run explanation methods and seamlessly evaluate the explanations. Similarly, our Dataset API leverages Hugging Face's *datasets*[5] to retrieve data and human rationales.

3 Case Studies

We showcase *ferret* in two real-world tasks, focusing on benchmarking explainers on individual samples or across multiple instances. In the following, we describe how *ferret* highlights the best explainers in sentiment analysis and hate speech detection tasks. Our running examples use an XLM-RoBERTa model fine-tuned for sentiment analysis (Barbieri et al., 2021) and a BERT model fine-tuned for hate speech detection (Mathew et al., 2021).

3.1 Faithfulness Metrics for Error Analysis

Explanations on individual instances are often used for model debugging and error analysis (Vig, 2019; Feng et al., 2018). However, different explanations can lead users to different conclusions, hindering a solid understanding of the model's flaws. We show how practitioners can alleviate this issue including *ferret* in their pipeline.

Figure 2 shows explanations and faithfulness metrics computed on the sentence "Great movie for a great nap!" for the "Positive" class label misclassified by the model as "Negative".

Faithfulness metrics show that SHAP adheres best to the model's inner workings since it returns the most comprehensive and relevant explanations. Indeed, SHAP retrieves the highest number of tokens the model used to make the prediction ($aopc\_compr(\uparrow) = 0.41$) that are relevant to drive the prediction ($aopc\_suff(\downarrow) = 0.09$). Further, $taucorr\_loo(\uparrow) = 0.43$ indicates that SHAP explanations capture the most important tokens for the prediction under the linearity assumption. Although Integrated Gradient (x Input) shows a higher $taucorr\_loo$, it does not provide comprehensive and sufficient explanations. Similarly, Gradient and Integrated Gradient show bad sufficiency and

[5] https://github.com/huggingface/datasets

```python
from transformers import
    AutoModelForSequenceClassification,
    AutoTokenizer
from ferret import Benchmark

name = "cardiffnlp/twitter-xlm-roberta-base
    -sentiment"
model = AutoModelForSequenceClassification.
    from_pretrained(name)
tokenizer = AutoTokenizer.
    from_pretrained(name)

bench = Benchmark(model, tokenizer)
query = "Great movie for a great nap!"

scores = bench.score(query)
print(scores)

# Run built-in explainers
explanations = bench.explain(
    query,
    target=2 # "Positive" label
)
bench.show_table(explanations)

# Evaluate explanations
evaluations = bench.evaluate_explanations(
    explanations, target=2
)
bench.show_evaluation_table(evaluations)

## Output
>> {'Negative': 0.013735532760620117,
>> 'Neutral': 0.06385018676519394,
>> 'Positive': 0.9224143028259277}
```

Token	_Great	_movie	_for	_a	_great	_nap	!
Partition SHAP	0.35	0.12	0.05	0.06	0.35	-0.00	0.05
LIME	-0.07	-0.08	0.03	-0.01	-0.24	0.17	0.06
Gradient	0.12	0.17	0.06	0.04	0.14	0.23	0.05
Gradient (x Input)	-0.11	-0.09	-0.08	0.03	0.03	0.11	-0.05
Integrated Gradient	-0.08	0.10	0.05	-0.06	0.00	0.03	-0.03
Integrated Gradient (x Input)	-0.09	-0.15	-0.17	-0.15	-0.10	-0.24	-0.10

	aopc_compr	aopc_suff	taucorr_loo
Partition SHAP	0.41	0.09	0.43
LIME	0.01	0.53	-0.33
Gradient	0.34	0.21	0.05
Gradient (x Input)	-0.01	0.44	-0.81
Integrated Gradient	0.05	0.50	-0.14
Integrated Gradient (x Input)	0.00	1.00	0.52

Figure 2: Code to explain and evaluate explanations on a sentiment classifier (top). Token attributions (middle): darker red (blue) show higher (lower) contribution to the prediction. Faithfulness metrics (bottom): darker colors show better performance.

comprehensiveness, respectively. LIME and Gradient (x Input) do not return trustworthy explanations according to all faithfulness metrics.

Once SHAP has been identified as the best explainer, its explanations enable researchers to investigate possible recurring patterns or detect model biases thoroughly. In this case, the explanations shed light on a type of lexical overfitting: the word "great" skews the prediction toward the positive label regardless of the context and semantics.

3.2 Multi-Instance Assessment

Instance-level analysis finds explainers that meet specific requirements locally. However, the best local explainer might be unsatisfactory across multiple instances. With *ferret*, users can easily produce and aggregate evaluation metrics across multiple dataset samples—or the entire corpus.

We describe how to choose the explainer that returns the most plausible and faithful explanations for the HateXplain dataset. For demonstration purposes, we focus only on a sample of the dataset.

Figure 3 (Appendix C) shows the metrics averaged across ten samples with the "hate speech" label. Results suggest again that SHAP yields the most faithful explanations. SHAP and Gradient achieve the best comprehensiveness and sufficiency scores, but SHAP outperforms all explainers for the τ correlation with LOO (*taucorr_loo* ($\uparrow$) = 0.41). Gradient provides the most plausible explanations, followed by SHAP.

4 Related Work

This section provides a review of tools and libraries that offer a subset of the *ferret*'s functionalities, namely the option to use multiple XAI methods and datasets, evaluation API, transformer-readiness, and built-in visualization. Table 2 summarizes them and compares *ferret* with similar frameworks.

Tools for Post-Hoc XAI. Toolkits for post-hoc interpretability offer built-in methods to explain model prediction, typically through a code interface. *ferret* builds on and extends this idea to a unified framework to generate explanations, *evaluate and compare* them, with support to several XAI datasets. Moreover, *ferret*'s explainers are integrated with transformers's (Wolf et al., 2020) principles and conventions.

PyTorch's Captum (Kokhlikyan et al., 2020) is a generic Python library supporting many interpretability methods. However, the library lacks integration with the Hugging Face Hub and offers no evaluation procedures. AllenNLP Interpret (Wallace et al., 2019b) provides interpretability methods based on gradients and adversarial attacks for AllenNLP models (Gardner et al., 2018). We borrow

	Multiple XAI approaches	Transformers-readiness	Evaluation APIs	XAI datasets	Built-in visualization
Captum	✓	✗	✗	✗	✓
AllenNLP Interpret	✓	✗	✗	✗	✗
Transformers-Interpret	✗	✓	✗	✗	✓
Thermostat	✓	✓	✗	✗	✓
ContrXT	✗	✗	✗	✗	✗
OpenXAI	✓	✗	✓	✗	✗
NLPVis	✗	✗	✗	✗	✓
Seq2Seq-Vis	✗	✗	✗	✗	✓
BertViz	✗	✓	✗	✗	✗
ELI5	✗	✗	✗	✗	✓
LIT	✓	✗	✗	✗	✓
ERASER	✗	✗	✓	✓	✗
Inseq	✓	✓	✗	✗	✓
ferret	✓	✓	✓	✓	✓

Table 2: Comparing off-the-shelf features across different XAI libraries. When assessing built-in visualization, we disregard tools that either do not provide a unified interface or provide single data-point visualizations.

the modular and extensible design and extend it to a wider set of explainers. Transformers-Interpret[6] leverages Captum to explain Transformer models, but it supports only a limited number of methods. Thermostat (Feldhus et al., 2021) exposes pre-computed feature attribution scores through the Hugging-Face Hub but no features oriented to implement or evaluate XAI. We support the Thermostat as a third-party add-on and let users test and benchmark pre-computed explanations. Unlike our study, Inseq (Sarti et al., 2023) focuses on post-hoc interpretability for sequence generation models. Although researchers can use the library to add interpretability evaluations to their models, the toolkit lacks built-in evaluation metrics.

Other related approaches enable global (rather than local) explainability (Malandri et al., 2022), or explanation interfaces for non-transformers models on non-NLP tasks (Agarwal et al., 2022). Other approaches study model behavior at the subgroup level (Wang et al., 2021; Goel et al., 2021; Pastor et al., 2021a,b), focusing more on model evaluation and robustness rather than its interpretation.

Visualization. Most studies that develop visualization tools to investigate the relationships among the input, the model, and the output focus either on specific NLP models - NLPVis (Liu et al., 2018), Seq2Seq-Vis (Strobelt et al., 2018), or explainers - BertViz (Vig, 2019), ELI5[7]. LIT (Tenney et al., 2020) streamlines exploration and analysis in different models. However, it acts mainly as a graphical browser interface. *ferret* provides a Python interface easy to integrate with pre-existing pipelines.

Evaluation. Although prior works introduced diagnostic properties for XAI techniques, evaluating them in practice remains challenging. Studies either concentrate on specific model architectures (Lertvittayakumjorn and Toni, 2019; Arras et al., 2019; DeYoung et al., 2020), individual datasets (Guan et al., 2019; Arras et al., 2019), or a single group of explainability methods (Robnik-Šikonja and Bohanec, 2018; Adebayo et al., 2018). Hence, providing a generally applicable and automated tool for choosing the most suitable method is crucial. To this end, Atanasova et al. (2020) present a comparative study of XAI techniques in three application tasks and model architectures. To the best of our knowledge, we are the first to present a user-friendly Python interface to interpret, visualize and empirically evaluate models directly from the Hugging Face Hub across several metrics. We extend previous work from DeYoung et al. (2020), who developed a benchmark for evaluating rationales on NLP models called ERASER by offering a unified interface for evaluation *and* visual comparison of the explanations at the instance- and dataset-level.

Closer to *ferret*, the OpenXAI framework (Agar-

[6] https://github.com/cdpierse/transformers-interpret

[7] https://github.com/TeamHG-Memex/eli5

wal et al., 2022) enables a systematic evaluation of feature attribution explanation, integrating multiple explainers and XAI structured datasets. OpenXAI supports tabular datasets while we focus on textual data and NLP models.

5 Conclusions

We introduced *ferret*, a novel Python framework to easily access XAI techniques on transformer models. With *ferret*, users can *explain* using state-of-the-art post-hoc explainability techniques, *evaluate* explanations on several metrics for faithfulness and plausibility, and easily *interact* with datasets annotated with human rationales.

We built *ferret* with modularity and abstraction in mind to facilitate future extensions and contributions from the community (see Appendix B for an overview of the ongoing development). As future work, we envision off-the-shelf support for new NLP tasks and scenarios. Building on the classification setup presented in this paper, we plan to add support to more NLP tasks that can be framed as classification, such as Mask Filling Prediction, Natural Language Inference, Zero-Shot Text Classification, Next Sentence Prediction, Token Classification, and Multiple-Choice QA. One further direction would be improving *ferret*'s interoperability with new libraries, e.g., Inseq (Sarti et al., 2023) for XAI on text generation tasks and models.

Ethics Statement

ferret's primary goal is to facilitate the comparison of methods that are instead frequently tested in isolation. Nonetheless, we cannot assume the metrics we currently implement provide a full, exhaustive picture, and we work towards enlarging this set accordingly.

Further, interpretability is much broader than post-hoc feature attribution. We focus on this family of approaches for their wide adoption and intuitiveness.

Similarly, the evaluation measures we integrate are based on removal-based criteria. Prior works pointed out their limitations, specifically the problem of erased inputs falling out of the model input distribution (Hooker et al., 2019).

Acknowledgments

This project has partially received funding from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation program (grant agreement No. 949944, INTEGRATOR), by Fondazione Cariplo (grant No. 2020-4288, MONICA), and by the grant "National Centre for HPC, Big Data and Quantum Computing", CN000013 (approved under the M42C Call for Proposals - Investment 1.4 - Notice "National Centers" - D.D. No. 3138, 16.12.2021, admitted for funding by MUR Decree No. 1031, 17.06.2022). DN and GA are members of the MilaNLP group and the Data and Marketing Insights Unit of the Bocconi Institute for Data Science and Analysis. EP did part of the work while at CENTAI and is currently a member of the DataBase and Data Mining Group (DBDMG) at Politecnico di Torino. CDB contributed to the work while at Bocconi University and is currently part of the UKRI Centre for Doctoral Training in Safe and Trusted Artificial Intelligence (www.safeandtrustedai.org).

References

Julius Adebayo, Justin Gilmer, Michael Muelly, Ian Goodfellow, Moritz Hardt, and Been Kim. 2018. Sanity checks for saliency maps. *Advances in neural information processing systems*, 31.

Chirag Agarwal, Satyapriya Krishna, Eshika Saxena, Martin Pawelczyk, Nari Johnson, Isha Puri, Marinka Zitnik, and Himabindu Lakkaraju. 2022. OpenXAI: Towards a transparent evaluation of model explanations. In *Thirty-sixth Conference on Neural Information Processing Systems Datasets and Benchmarks Track*.

Leila Arras, Ahmed Osman, Klaus-Robert Müller, and Wojciech Samek. 2019. Evaluating recurrent neural network explanations. In *Proceedings of the 2019 ACL Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 113–126, Florence, Italy. Association for Computational Linguistics.

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, Online. Association for Computational Linguistics.

Giuseppe Attanasio, Debora Nozza, Eliana Pastor, and Dirk Hovy. 2022. Benchmarking post-hoc interpretability approaches for transformer-based misogyny detection. In *Proceedings of NLP Power! The First Workshop on Efficient Benchmarking in NLP*, pages 100–112, Dublin, Ireland. Association for Computational Linguistics.

Francesco Barbieri, Luis Espinosa Anke, and José Camacho-Collados. 2021. XLM-T: A multilingual language model toolkit for twitter. *CoRR*, abs/2104.12250.

Samuel Carton, Anirudh Rathore, and Chenhao Tan. 2020. Evaluating and characterizing human rationales. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 9294–9307, Online. Association for Computational Linguistics.

Marina Danilevsky, Kun Qian, Ranit Aharonov, Yannis Katsis, Ban Kawas, and Prithviraj Sen. 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*, pages 447–459, Suzhou, China. Association for Computational Linguistics.

Jay DeYoung, Sarthak Jain, Nazneen Fatema Rajani, Eric Lehman, Caiming Xiong, Richard Socher, and Byron C. Wallace. 2020. ERASER: A benchmark to evaluate rationalized NLP models. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4443–4458, Online. Association for Computational Linguistics.

Nils Feldhus, Robert Schwarzenberg, and Sebastian Möller. 2021. Thermostat: A large collection of NLP model explanations and analysis tools. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 87–95, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Shi Feng, Eric Wallace, Alvin Grissom II, Mohit Iyyer, Pedro Rodriguez, and Jordan Boyd-Graber. 2018. Pathologies of neural models make interpretations difficult. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3719–3728, Brussels, Belgium. Association for Computational Linguistics.

Matt Gardner, Joel Grus, Mark Neumann, Oyvind Tafjord, Pradeep Dasigi, Nelson F. Liu, Matthew Peters, Michael Schmitz, and Luke Zettlemoyer. 2018. AllenNLP: A deep semantic natural language processing platform. In *Proceedings of Workshop for NLP Open Source Software (NLP-OSS)*, pages 1–6, Melbourne, Australia. Association for Computational Linguistics.

Karan Goel, Nazneen Fatema Rajani, Jesse Vig, Zachary Taschdjian, Mohit Bansal, and Christopher Ré. 2021. Robustness gym: Unifying the NLP evaluation landscape. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies: Demonstrations*, pages 42–55, Online. Association for Computational Linguistics.

Chaoyu Guan, Xiting Wang, Quanshi Zhang, Runjin Chen, Di He, and Xing Xie. 2019. Towards a deep and unified understanding of deep neural models in nlp. In *International conference on machine learning*, pages 2454–2463. PMLR.

Sara Hooker, Dumitru Erhan, Pieter-Jan Kindermans, and Been Kim. 2019. A benchmark for interpretability methods in deep neural networks. In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc.

Alon Jacovi and Yoav Goldberg. 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*, pages 4198–4205, Online. Association for Computational Linguistics.

Sarthak Jain and Byron C. Wallace. 2019. Attention is not Explanation. 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 3543–3556, Minneapolis, Minnesota. Association for Computational Linguistics.

Xisen Jin, Zhongyu Wei, Junyi Du, Xiangyang Xue, and Xiang Ren. 2019. Towards hierarchical importance attribution: Explaining compositional semantics for neural sequence models. In *International Conference on Learning Representations*.

Narine Kokhlikyan, Vivek Miglani, Miguel Martin, Edward Wang, Bilal Alsallakh, Jonathan Reynolds, Alexander Melnikov, Natalia Kliushkina, Carlos Araya, Siqi Yan, et al. 2020. Captum: A unified and generic model interpretability library for pytorch. *arXiv preprint arXiv:2009.07896*.

Piyawat Lertvittayakumjorn and Francesca Toni. 2019. Human-grounded evaluations of explanation methods for 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 5195–5205, Hong Kong, China. Association for Computational Linguistics.

Jiwei Li, Will Monroe, and Dan Jurafsky. 2016. Understanding neural networks through representation erasure. *CoRR*, abs/1612.08220.

Shusen Liu, Tao Li, Zhimin Li, Vivek Srikumar, Valerio Pascucci, and Peer-Timo Bremer. 2018. Visual interrogation of attention-based models for natural language inference and machine comprehension. Technical report, Lawrence Livermore National Lab.(LLNL), Livermore, CA (United States).

Scott M Lundberg and Su-In Lee. 2017. A unified approach to interpreting model predictions. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

Andreas Madsen, Siva Reddy, and Sarath Chandar. 2022. Post-hoc interpretability for neural nlp: A survey. *ACM Comput. Surv.* Just Accepted.

Lorenzo Malandri, Fabio Mercorio, Mario Mezzanzanica, Navid Nobani, and Andrea Seveso. 2022. Contrastive explanations of text classifiers as a service. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies: System Demonstrations*, pages 46–53.

Binny Mathew, Punyajoy Saha, Seid Muhie Yimam, Chris Biemann, Pawan Goyal, and Animesh Mukherjee. 2021. Hatexplain: A benchmark dataset for explainable hate speech detection. *Proceedings of the AAAI Conference on Artificial Intelligence*, 35(17):14867–14875.

Eliana Pastor and Elena Baralis. 2019. Explaining black box models by means of local rules. In *Proceedings of the 34th ACM/SIGAPP Symposium on Applied Computing*, SAC '19, page 510–517, New York, NY, USA. Association for Computing Machinery.

Eliana Pastor, Luca de Alfaro, and Elena Baralis. 2021a. Looking for trouble: Analyzing classifier behavior via pattern divergence. In *Proceedings of the 2021 International Conference on Management of Data*, SIGMOD '21, page 1400–1412, New York, NY, USA. Association for Computing Machinery.

Eliana Pastor, Andrew Gavgavian, Elena Baralis, and Luca de Alfaro. 2021b. How divergent is your data? *Proc. VLDB Endow.*, 14(12):2835–2838.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, Peter J Liu, et al. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *J. Mach. Learn. Res.*, 21(140):1–67.

Marco Túlio 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, San Francisco, CA, USA, August 13-17, 2016*, pages 1135–1144. ACM.

Marko Robnik-Šikonja and Marko Bohanec. 2018. Perturbation-based explanations of prediction models. In *Human and machine learning*, pages 159–175. Springer.

Alexis Ross, Ana Marasović, and Matthew Peters. 2021. Explaining NLP models via minimal contrastive editing (MiCE). In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 3840–3852, Online. Association for Computational Linguistics.

Soumya Sanyal and Xiang Ren. 2021. Discretized integrated gradients for explaining language models. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 10285–10299, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Gabriele Sarti, Nils Feldhus, Ludwig Sickert, and Oskar van der Wal. 2023. Inseq: An interpretability toolkit for sequence generation models. *ArXiv*, abs/2302.13942.

Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. 2014a. Deep inside convolutional networks: Visualising image classification models and saliency maps. In *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Workshop Track Proceedings*.

Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. 2014b. 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 Y 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.

Hendrik Strobelt, Sebastian Gehrmann, Michael Behrisch, Adam Perer, Hanspeter Pfister, and Alexander M Rush. 2018. S eq 2s eq-v is: A visual debugging tool for sequence-to-sequence models. *IEEE transactions on visualization and computer graphics*, 25(1):353–363.

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*, ICML'17, page 3319–3328. JMLR.org.

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.

Jesse Vig. 2019. Visualizing attention in transformer-based language representation models. *arXiv preprint arXiv:1904.02679*.

Eric Wallace, Shi Feng, Nikhil Kandpal, Matt Gardner, and Sameer Singh. 2019a. Universal adversarial triggers for attacking and analyzing NLP. 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 2153–2162, Hong Kong, China. Association for Computational Linguistics.

Eric Wallace, Jens Tuyls, Junlin Wang, Sanjay Subramanian, Matt Gardner, and Sameer Singh. 2019b. 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.

Xiao Wang, Qin Liu, Tao Gui, Qi Zhang, Yicheng Zou, Xin Zhou, Jiacheng Ye, Yongxin Zhang, Rui Zheng, Zexiong Pang, Qinzhuo Wu, Zhengyan Li, Chong Zhang, Ruotian Ma, Zichu Fei, Ruijian Cai, Jun Zhao, Xingwu Hu, Zhiheng Yan, Yiding Tan, Yuan Hu, Qiyuan Bian, Zhihua Liu, Shan Qin, Bolin Zhu, Xiaoyu Xing, Jinlan Fu, Yue Zhang, Minlong Peng, Xiaoqing Zheng, Yaqian Zhou, Zhongyu Wei, Xipeng Qiu, and Xuanjing Huang. 2021. TextFlint: Unified multilingual robustness evaluation toolkit for natural language processing. 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 347–355, Online. Association for Computational Linguistics.

Sarah Wiegreffe and Ana Marasovic. 2021. Teach me to explain: A review of datasets for explainable natural language processing. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 1)*.

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.

Fan Yin, Zhouxing Shi, Cho-Jui Hsieh, and Kai-Wei Chang. 2022. On the sensitivity and stability of model interpretations in NLP. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2631–2647, Dublin, Ireland. Association for Computational Linguistics.

Omar Zaidan and Jason Eisner. 2008. Modeling annotators: A generative approach to learning from annotator rationales. In *Proceedings of the 2008 conference on Empirical methods in natural language processing*, pages 31–40.

A Technical Details

A.1 Explainer API

Our implementation is built on top of original implementations (as for SHAP and LIME) and open-source libraries (as Captum (Kokhlikyan et al., 2020) for gradient-based explainers) to directly explain Transformer-based language models.

Currently, we integrate Gradient (G) (Simonyan et al., 2014b), Integrated Gradient (IG) (Sundararajan et al., 2017), SHAP (Lundberg and Lee, 2017), and LIME (Ribeiro et al., 2016). For G and IG, users can get explanations from plain gradients or multiply gradients by the input token embeddings. For SHAP, we use the Partition approximation to estimate Shapley values.[8]

A.1.1 Evaluation API

While human gold annotations are normally discrete, current explainers provide continuous token attribution scores. Following previous work, we hence go from continuous scores to a discrete set of *relevant* tokens (i.e., r_j in Section 2.3) as follows.

We consider only tokens with a positive contribution to the chosen label (i.e., they *push* the prediction towards the chosen label). For the AOPC comprehensiveness and sufficiency measures, the relevant tokens in the discrete rationale are the most $k\%$ important tokens with k ranging from 10% to 100% (step of 10%). For token-level IOU and F1 scores plausibility measure, we follow the DeYoung et al. (2020) and Mathew et al. (2021) approach, and we select the top k tokens where k is the average length of human rationales for the dataset.

The evaluation measures at the dataset level are the average scores across explanations. Differently than DeYoung et al. (2020) that use the F1 IOU score, we directly compute the average token-level IOU.

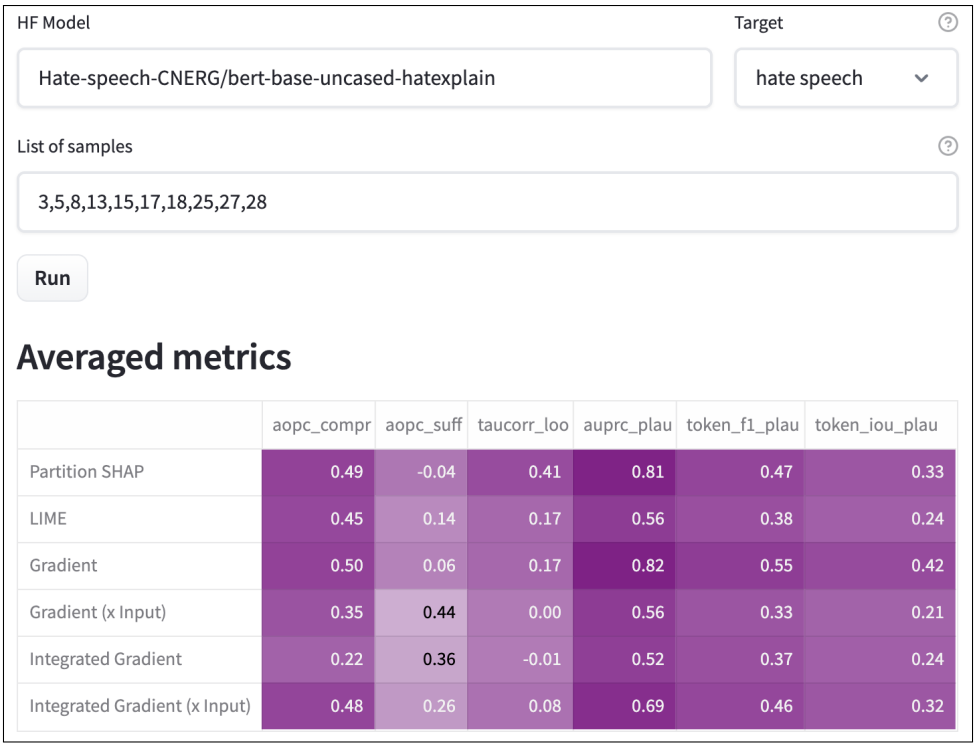
All human rationales are at the token level, indicating the most relevant tokens to a given class label.

A.2 Technical Features

ferret implements several functionalities to facilitate end users in using it.

- High-level interface. Most of *ferret*'s features, such as interpretability methods and evalua-

[8] https://shap.readthedocs.io/en/latest/generated/shap.explainers.Partition.html



	aopc_compr	aopc_suff	taucorr_loo	auprc_plau	token_f1_plau	token_iou_plau
Partition SHAP	0.49	-0.04	0.41	0.81	0.47	0.33
LIME	0.45	0.14	0.17	0.56	0.38	0.24
Gradient	0.50	0.06	0.17	0.82	0.55	0.42
Gradient (x Input)	0.35	0.44	0.00	0.56	0.33	0.21
Integrated Gradient	0.22	0.36	-0.01	0.52	0.37	0.24
Integrated Gradient (x Input)	0.48	0.26	0.08	0.69	0.46	0.32

Figure 3: Faithfulness and Plausibility metrics averaged across ten samples with the "hateful" label of HateXPlain. Darker colors mean better performance.

tion measures, are accessible via a single entry point, the `Benchmark` class.

- GPU-enabled batched inference. *ferret* requires running inference for certain executions. It uses batching and local GPUs transparently to the user whenever that happens.

- Visualization methods. The `Benchmark` class exposes several methods to visualize attribution scores and evaluation results in tabular format. These tables are plotted seamlessly on Jupyter Notebooks (see Figure 2 (bottom) for an example).

B Ongoing Development

ferret is under active development. We are extending the core modules as follows.

Explainers. We plan to integrate two recent interpretability methods that require training a complementary model. Sampling and Occlusion (SOC) (Jin et al., 2019) provides a hierarchical explanation to address compositional contributions. Minimal Contrastive Editing (MiCE) (Ross et al., 2021) trains a T5 (Raffel et al., 2020) model to implement contrastive edits to the input to change the model output. Finally, we are including a third gradient-based algorithm. Integrated Discretized Gradients (Sanyal and Ren, 2021) improve IG sampling intermediate steps close to actual words in the embedding space.

Evaluators. We plan to include additional evaluation measures such as sensitivity, stability (Yin et al., 2022), and Area Under the Threshold-Performance curve (AUC-TP) (Atanasova et al., 2020).

C Additional Results

Figure 3 shows a screenshot of dataset-level assessment from our demo web app. It reports the evaluation metrics averaged across ten samples with the "hate speech" label for the HateXplain dataset, discussed in Section 3.

The user specifies a model from the Hugging Face Hub (*HF Model* field), the target class (*Target*), and the set of samples of interest (*List of samples*). *ferret* web app directly computes explanation and their evaluation and visualizes the results.