Zero-shot Code-Mixed Offensive Span Identification through Rationale Extraction

Manikandan Ravikiran[†*], **Bharathi Raja Chakravarthi**[‡]

[†]Georgia Institute of Technology, Atlanta, Georgia
[‡]Data Science Institute, National University of Ireland Galway
mravikiran3@gatech.edu, bharathi.raja@insight-centre.org

Abstract

This paper investigates the effectiveness of sentence-level transformers for zero-shot offensive span identification on a code-mixed Tamil dataset. More specifically, we evaluate rationale extraction methods of Local Interpretable Model Agnostic Explanations (LIME) (Ribeiro et al., 2016a) and Integrated Gradients (IG) (Sundararajan et al., 2017) for adapting transformer based offensive language classification models for zero-shot offensive span identification. To this end, we find that LIME and IG show baseline F_1 of 26.35% and 44.83%, respectively. Besides, we study the effect of data set size and training process on the overall accuracy of span identification. As a result, we find both LIME and IG to show significant improvement with Masked Data Augmentation and Multilabel Training, with F_1 of 50.23% and 47.38% respectively. *Disclaimer : This paper contains examples that may be considered profane, vulgar, or offensive. The examples do not represent the views of the authors or their employers/graduate schools towards any person(s), group(s), practice(s), or entity/entities. Instead they are used to emphasize only the linguistic research challenges.*

1 Introduction

Offensive language classification and offensive span identification from code-mixed Tamil-English comments portray the same task at different granularities. In the former case, we classify if the code mixed sentence is offensive or not, while the latter concentrates on extracting the offensive parts of the comments. Accordingly, one could do the former using models of the latter and vice versa. Transformer-based architectures such as BERT (Devlin et al., 2019), RoBERTa (Liu et al., 2019) and XLM-RoBERTa (Conneau et al., 2020) have achieved state-of-the-art results on both these tasks (Chakravarthi et al., 2021a). However, often

these tasks are treated as independent and model development is often separated.

This paper studies the rationale extraction methods for inferring offensive spans from transformer models trained only on comment-level offensive language classification labels. Such an idea is often vital in the case of code-mixed Tamil-English comments for which span annotations are often costly to obtain, but comment-level labels are readily available. Besides, such an approach will also help in decoding the models' logic behind the prediction of offensiveness.

Accordingly, we evaluate and compare two different methods, namely LIME and IG, for adapting pre-trained transformer models into zero-shot offensive span labelers. Our experiments show that using LIME with pre-trained transformer models struggles to infer correct span level annotations in a zero-shot manner, achieving only 20% F_1 on offensive span identification for code-mixed Tamil-English comments. To this end, we find that a combination of masked data augmentation and multilabel training of sentence transformers helps to better focus on individual necessary tokens and achieve a strong baseline on offensive span identification. Besides, IG consistently surpasses LIME even in cases where there are no data augmentation or multilabel training. Overall the contributions of this paper are as follows.

- We introduce preliminary experiments on offensive language classification transformer models for zero-shot offensive span identification from code-mixed Tamil-English language comments.

- We systematically compare LIME and IG methods for zero-shot offensive span identification.

- We study the impact of data and training process on offensive span identification by

---

[*]Corresponding Author

proposing masked data augmentation and multilabel training.

- We further release our code, models, and data to facilitate further research in the field[1].

The rest of the paper is organized as follows. In Section 2, we present LIME and IG methods in brief. Meanwhile section 3 and 4 focus on dataset and experimental setup. In section 5, we present detailed experiments and conclude in section 6 with our findings and possible implications on the future work.

2 Methods

In this section, we present the two rationale extraction methods LIME and IG used to turn sentence-level transformer models into zero-shot offensive span labelers.

2.1 Local Interpretable Model Agnostic Explanation (LIME)

LIME (Ribeiro et al., 2016b) is a model agnostic interpretability approach that generates word-level attribution scores using local surrogate models that are trained on perturbed sentences generated by randomly masking out words in the input sentence. The LIME model has seen considerable traction in the context of rationale extraction for text classification, including work by Thorne et al. (2019), which suggests that LIME outperforms attention-based approaches to explain NLI models. LIME was also used to probe an LSTM based sentence-pair classifier (Lan and Xu, 2018) by removing tokens from the premise and hypothesis sentences separately. The generated scores are used to perform binary classification of tokens, with the threshold based on F_1 performance on the development set. The token-level predictions were evaluated against human explanations of the entailment relation using the e-SNLI dataset (Camburu et al., 2018).

Meanwhile, for offensive span identification in English Ding and Jurgens (2021) coupled LIME with RoBERTa trained on an expanded training set to find expanded training set could help RoBERTa more accurately learn to recognize toxic span. However, though LIME outperforms other methods, it is significantly slower than Integrated Gradients methods, presented in the next section.

---

[1]The code and data is made available at `https://github.com/manikandan-ravikiran/zero-shot-offensive-span`

2.2 Integrated Gradients (IG)

Integrated Gradients (Sundararajan et al., 2017) focuses on explaining predictions by integrating the gradient along some trajectory in input space connecting two points. Integrated gradient and its variants are widely used in different fields of deep learning including natural language processing (Sikdar et al., 2021).

Specifically, it is an iterative method, which starts with so-called starting baselines, i.e., a starting point that does not contain any information for the model prediction. In our case involving textual data, this is the set exclusively with the start and end tokens. These tokens mark the beginning and the end of a sentence and do not give any information about whether the evaluation is offensive or not. Following this, it takes a certain number of iterations, where the model moves from the starting baseline to the actual input to the model.

This iterative improvement approach is analogous to the sentence creation process wherein each step, we create the sentence word by word and calculate the offensiveness, which in turn gives us the attribution of the input feature. Across its iterations, whenever IG includes an offensive word, we can expect offensive classification prediction to swing more towards offensive class and vice versa. Such behavior will help calculate the attribution of each word in the identified sentence.

3 Datasets

In this section, we present various datasets used in this study. Details on how they are used across different experiments are presented in Table 3. Finally, the overall dataset statistics are as shown in Table 1.

3.1 Offensive Span Identification Dataset

The Shared task on Offensive Span Identification from Code-Mixed Tamil English Comments (Ravikiran and Annamalai, 2021) focuses on the extraction of offensive spans from Youtube Comments. The dataset contains 4786 and 876 examples across its train and test set respectively. It consists of annotated offensive spans indicating character offsets of parts of the comments that were offensive.

3.2 Masked Augmented Dataset

The data available from both Ravikiran and Annamalai (2021) is minimal, and transformer methods

Dataset	Number of Train Samples	Number of Test Samples	Offensive Language Classification	Offensive Span Identification
Offensive Span Identification Dataset	4786	876	✗	✓
Mask Augmented Dataset	109961	-	✓	✓
Multilabel Dataset	109961	-	✓	✓

Table 1: Dataset statistics of various datasets used in this work.

Steps	Example Output
Offensive Lexicon Creation	[sanghu, thu,thooooo,suthamm,F**k,flop,w*f, p**a,n**y,nakkal]
Data Sourcing	Last scene vera level love u [Last, scene, vera, level, love,u]
Mask Generation	[0,1, 0, 0, 1,1] [1,0, 1, 1, 1,1] [0,0, 0, 0, 1,1]
Offensive Word Augmentation	[Last, sangh, vera, level, thu, n**y] [flop, scene, thooooo, suthamm, F**k, sanghu] [Last, scene, p**a, n**y, love, u]
Position Identification and Multilabel Creation	Last sangh vera level thu n**y [1,0,1] flop scene thooooo suthamm F**k sanghu [1,1,1] Last scene p**a n**y love u [0,1,0]

Table 2: Various steps in Masked Augmented Dataset and Multilabel Dataset creation with sample outputs.

are sensitive to dataset size (Xu et al., 2021). Thus we created an additional dataset using Masked Augmentation. Accordingly, the data is generated by using the following steps.

- **Step 1: Offensive Lexicon Creation**: First, we create an offensive lexicon from the train set of offensive span identification datasets. To do this, we do following

  - Extract the phrases corresponding to annotated offensive spans from the training dataset of Ravikiran and Annamalai (2021).
  - Selecting phrases of size less than 20 characters and word tokenizing them to extract the individual words.
  - Manually, post-processing these words to ignore words that are not offensive. For example, many phrases include conjunctions and pronouns which are not directly offensive.

Accordingly, an offensive lexicon with 2900 tokens is created.

- **Step 2: Data Sourcing**: In this step, we select the dataset used for creating the Masked Augmented dataset. Specifically, we use the 25425 non-offensive comments from Dravidian Code-Mix dataset (Chakravarthi et al., 2021b).

- **Step 3: Mask Generation**: The mask generation is done as follows

  - Each of 25425 non-offensive comments was tokenized to create respective *maskable token list*.
  - Three random binary masks are generated for each of the tokenized non-offensive comments. These binary masks have same length as that of its maskable token list.

- **Step 4: Offensive Word Augmentation:** Finally, words with a corresponding binary mask of 1 are replaced with words randomly selected from the offensive lexicon from step 1. Additionally, the spans corresponding to the words that were replaced are saved.

Overall, such augmentation resulted in 109961 comments, with 75009 being offensive comments and 34952 non-offensive comments. Table 2 shows an example sentence and masked augmented dataset creation process.

3.3 Multilabel Dataset

All the previously mentioned datasets are restricted to classification only i.e. they contain a binary label indicating if they are offensive or they have annotated offensive spans. Additionally, these sentences does not explicitly encode any position information of the offensive words, which is useful for

242

Experiment Name	Train dataset	Test dataset
Benchmark	Offensive Span Identification Dataset (Ravikiran and Annamalai, 2021; Ravikiran et al., 2022)	
OS-Baseline	Dravidian CodeMix (Chakravarthi et al., 2021b)	Offensive Span
OS-Augmentation	Mask Augmented Dataset	Identification
OS-Multilabel	Multilabel Dataset	Dataset

Table 3: Relationship between the datasets and experiments.

Name	Value
γ	0.1
max seq length	150
train batch size	64
eval batch size	64
warmup ratio	0.1
learning rate	$3x10^{-5}$
weight decay	0.1
initializer	glorot

Table 4: Model Hyperparameters for Training Transformers.

training. Work of Ke et al. (2021) show encoding relative positional information based attention directly in each head often improves the overall result of corresponding down stream task. Similarly Shaw et al. (2018) also proposed using relative position encoding instead of absolute position encoding and couple them with key and value projections of Transformers to improve overall results. As such, in this work, to encode position we create a multilabel dataset in which the labels indicate the relative position of offensive words. The multilabel dataset is created as follows.

- **Step 1: Dataset Selection:** We first select the 109961 comments from the Masked Augmented Dataset along with their saved spans.

- **Step 2: Position Identification and Multilabel Creation:** From the identified spans, we check if the offensive spans are present in (a) start of the comment (b) end of the comment and (c) middle of the comment. Depending on presence of offensiveness we create three binary labels. For example in Table 2 for sentence `Last scene p**a n**y love u` we can see that the offensive word to be present in the center of the sentence. Accordingly we give it a label [0,1,0]. Meanwhile for comment `Last sangh vera level thu n**y` we can see offensive words to be in center and at the end thus we give label [1,0,1].

4 Experimental Setup

In this section, we present our experimental setup in detail. All of our experiments follow the two steps as explained below.

Transformer Training: We use three different transformer models, namely Multilingual-BERT, RoBERTa, and XLM-RoBERTa, made available by Hugging Face (Wolf et al., 2019), as our transformer architecture due to their widespread usage in the context of code-mixed Tamil-English Offensive Language Identification. In line with the works of Mosbach et al. (2021) all the models were fine-tuned for 20 epochs, and the best performing checkpoint was selected. Each transformer model takes 1 hour to train on a Tesla-V100 GPU with a learning rate of $3x10^{-5}$. Further, all of our experiments were run five times with different random seeds and the results so reported are an average of five runs. The relationship between the datasets used to train the transformers across various experiments is as shown in Table 3. Meanwhile, the model hyperparameters are presented in Table 4.

Span Extraction Testing: After training the transformer models for offensive language identification, we use the test set from the offensive span identification dataset for testing purposes. For LIME, we use individual transformer models' MASK token to mask out individual words and allow LIME to generate 5000 masked samples per sentence. The resulting explanation weights are then used as scores for each word, and tokens below the fixed decision threshold of $\tau = -0.01$ are removed while the spans of the rest of the comments are used for offensive span identification. Meanwhile, for the IG model, for each sentence in the test set, we perform 50 iterations to generate scores for each word and extract the spans in line with LIME.

5 Experiments, Results and Analysis

The consolidated results are presented in Table 5. Each model is trained as an offensive comment classifier and then evaluated for offensive span identification. Though we do not explicitly furnish any

Experiments	Model	$F_1(\%)$	
		LIME	IG
Benchmark	BENCHMARK 1	39.834	
	BENCHMARK 2	37.024	
OS-Baseline	BERT	26.35	44.83
	RoBERTa	24.26	37.01
	XLM-RoBERTa	22.86	43.13
OS-Augmentation	BERT	24.97	44.83
	RoBERTa	26.23	44.98
	XLM-RoBERTa	21.93	**50.23**
OS-Multilabel	BERT	47.137	44.83
	RoBERTa	**47.38**	35.83
	XLM-RoBERTa	46.76	42.06

Table 5: Consolidated Results on Offensive Span Identification Dataset. All the values represent character level F_1 measure.

signals regarding which words are offensive, we can see an assortment of behaviors across both the rationale extraction methods when trained differently. For reference comparison, we also include two benchmark baseline models from Ravikiran et al. (2022). BENCHMARK 1 is a random baseline model which haphazardly labels 50% of characters in comments to belong to be offensive inline. BENCHMARK 2 is a lexicon-based system, which first extracted all the offensive words from the train samples of offensive span identification dataset (Ravikiran and Annamalai, 2021). These words were scoured in comments from the test set during inference, and corresponding spans were noted. We report the character level F_1 for extracted spans inline with Ravikiran et al. (2022).

5.1 OS-Baseline Experiments

Firstly, both benchmarks exhibit high performance, making the task competitive for LIME and IG methods. To start with, we analyze the results of OS-Baseline experiments. From Table 5, we can see that LIME has moderately low performance compared to IG, which either beats the baseline or produces very close results. Analogizing the LIME and IG, we can see that IG has an average difference of 18% compared to LIME. To understand this, we identify various examples (Table 7) where LIME fails, and IG performs significantly well and vice versa. Firstly, we can see that LIME explicitly focuses on identifying overtly offensive words only. Besides, we can also see LIME focuses primarily on offensive words, while IG accounts for terms such as "Dei", "understood", "iruku poliye" etc.

Accordingly, to comprehend their performance

on offensive comments of different sizes, we separate results across (a) comments with less than 30 characters (F_1@30), (b) comments with 30-50 characters (F_1@50) (c) comments with more than 50 characters (F_1@>50). The results so obtained are as shown in Table 6. Accordingly, we find interesting outcomes. Firstly we can see that though LIME has lower F_1 overall, it tends to show competitive results against IG for comments with less than 30 characters.

With the increase in the comment length, the performance of LIME tends to lower considerably. We believe such behavior of LIME could be because of two reasons (a) surrogate models may not be strong enough to distinguish different classes and (b) dilution of scores due to LIME's random perturbation procedure. With random perturbations, the instances generated may be quite different from training instances drawn from the underlying distribution. Meanwhile, IG is compatible across all the sizes, and in the case of comments with less than 30 and 50 characters, we can see IG to show the result as high as 50%.

5.2 OS-Augmentation Experiments

Since transformers are very sensitive to dataset size, we focus on estimating the impact of dataset size used to train the transformers for offensive comment classification on the performance of LIME and IG, respectively. To this end, we used the Mask Augmented dataset to finetune the transformers and pose the question *Does adding data make any difference?* The various result so obtained are as shown in Table 5. Firstly, for LIME, we see no such drastic difference in F_1. However, for IG, we can see a significant improvement, especially for RoBERTa and XLM-RoBERTa models. Specifically, we can see the XLM-RoBERTa model to reach an accuracy of 50.23% with an average of 12% higher results compared to benchmark models and 7% compared to OS-Baseline.

Furthermore, analysis of results shows a couple of fascinating characteristics for XLM-RoBERTa. Firstly, we could see many predictions concentrating on words part of the long offensive span annotations. We believe this is because of the ability of the model to learn relations between words in different languages as part of its pretraining, which is not the case with M-BERT and RoBERTa. To verify this again, we separate the results across different comment sizes. From Table 6 we can see that

Experiments	Model	$F_1@30$ (%)		$F_1@50$ (%)		$F_1@>50$ (%)	
		LIME	IG	LIME	IG	LIME	IG
OS-Baseline	**BERT**	47.02	45.79	32.54	50.62	23.27	42.48
	RoBERTa	37.35	36.42	32.75	42.04	20.56	34.95
	XLM-RoBERTa	43.05	51.7	31.54	48.69	18.63	40.49
OS-Augmentation	**BERT**	48.45	45.79	32.62	50.62	21.29	42.48
	RoBERTa	50.21	45.86	32.71	50.71	22.8	42.66
	XLM-RoBERTa	31.19	59.47	27.58	57.01	19.17	47.19
OS-Multilabel	**BERT**	45.7	45.79	57.15	50.62	42.722	42.48
	RoBERTa	58.66	45.86	57.19	50.71	43	42.66
	XLM-RoBERTa	**59.84**	**59.47**	**57.62**	**57.01**	**42.95**	**47.19**

Table 6: Results across different size of comments.

Category	Comment Type	Examples
Correct Prediction	Comments with less than 30 characters	Dei like poda anaithu 9 p***a Semma mokka and as usual a masala movie
	Comments with 30-50 characters	M***u adichutu sagunga da j***i p**********a 81k views 89k likes YouTube be like W*F
	Comments with greater than 50 characters	Old vijayakanth movie parthathu pola irruku..pidikala.... Dei Yappa munjha paarthu Sirichu Sirichu vayiru vazhikuthu
Incorrect Prediction	Comments with less than 30 characters	except last scene its a crap Movie is going to be disaster
	Comments with 30-50 characters	Kandasamy and Mugamoodi mixed nu nenaikre.... Last la psycho ilayaraja nu solitan
	Comments with greater than 50 characters	All I understood from this video was Vikram likes Dosai.. Padam nichiyam oodama poga neriya vaipu iruku poliye ! Oru dislike ah potu vaipom

Table 7: Example of correct and incorrect predictions. Blue highlight shows words attributed by LIME. Green highlight shows words attributed by IG. Pink highlight shows words attributed by both LIME and IG. Yellow highlight shows parts of comments annotated in ground truth but not identified by both LIME and IG.

for longer sized comments, the model tends to outperform M-BERT, RoBERTa when coupled with IG. Meanwhile, LIME has no changes irrespective of used transformers.

5.3 OS-Multilabel Experiments

Finally, we analyze the significance of encoding the position of offensive words as part of the training process. To this end, we ask *Does introducing position information as part of the training process improve zero-shot results?*. As such, we use the multilabel dataset to finetune the transformers to obtain results, as shown in Table 5. Firstly, we can see that introducing multiple labels for training has no impact on the overall results of LG. However, we can see that LIME demonstrates a significant gain in overall results. Specifically, with multilabel training, the baseline results improve by 20% to 47.38%.

Furthermore, we can observe an equivalent trend across the different sizes of comments as seen in Table 6. In fact, for words of less than 30 and 50 characters, LIME outdoes IG models, which aligns with our hypothesis that the position is helpful. Overall from all the results from Table 5-6 we can see XLM-RoBERTa be more suitable for extracting

spans, especially with the addition of more data and position information. Meanwhile, IG is consistent in producing explanations irrespective of dataset size or training approach.

6 Conclusion

This work examines rationale extraction methods for inferring offensive spans from the transformer model trained for offensive sentence classification. Experiments revealed that approaches such as LIME do not perform as well when applied to transformers directly, attributing to potential issues with surrogate models and perturbation procedures. Meanwhile, we can see IG as the clear front runner for identifying offensive spans in a zero-shot way. We think this is due to the inherent nature of the method, where it focuses on creating the input at the same time learning the reason for offensiveness.

Besides, we also analyzed LIME and IG under large datasets and incorporated position information in the training process. To this end, we discovered that only augmenting does not improve the performance of LIME. However, when this large data is coupled with labels incorporating position information, both LG and IG improve significantly. Especially LIME prefers this approach with large

improvements on F_1, despite IG outperforming LIME.

Additionally, we also found XLM-RoBERTa to be a clear winner among the transformer models owing to its intrinsic learning of relationships which potentially helps with comments that are longer size. However, many details were unexplored, including (i) the effect of random perturbations on overall results (ii) the approach to merge attributions of multilabel predictions, which we plan to explore in the immediate future.

Acknowledgements

References

Oana-Maria Camburu, Tim Rocktäschel, Thomas Lukasiewicz, and Phil Blunsom. 2018. e-snli: Natural language inference with natural language explanations. *CoRR*, abs/1812.01193.

Bharathi Raja Chakravarthi, Ruba Priyadharshini, Navya Jose, Anand Kumar M, Thomas Mandl, Prasanna Kumar Kumaresan, Rahul Ponnusamy, Hariharan R L, John P. McCrae, and Elizabeth Sherly. 2021a. Findings of the shared task on offensive language identification in Tamil, Malayalam, and Kannada. In *Proceedings of the First Workshop on Speech and Language Technologies for Dravidian Languages*, pages 133–145, Kyiv. Association for Computational Linguistics.

Bharathi Raja Chakravarthi, Ruba Priyadharshini, Vigneshwaran Muralidaran, Navya Jose, Shardul Suryawanshi, Elizabeth Sherly, and John P. McCrae. 2021b. Dravidiancodemix: Sentiment analysis and offensive language identification dataset for dravidian languages in code-mixed text. *CoRR*, abs/2106.09460.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Unsupervised cross-lingual representation learning at scale. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, pages 8440–8451. Association for Computational Linguistics.

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, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pages 4171–4186. Association for Computational Linguistics.

Huiyang Ding and David Jurgens. 2021. HamiltonDinggg at SemEval-2021 task 5: Investigating toxic span detection using RoBERTa pre-training. In *Proceedings of the 15th International Workshop on Semantic Evaluation (SemEval-2021)*, pages 263–269, Online. Association for Computational Linguistics.

Guolin Ke, Di He, and Tie-Yan Liu. 2021. Rethinking positional encoding in language pre-training. *ArXiv*, abs/2006.15595.

Wuwei Lan and Wei Xu. 2018. Neural network models for paraphrase identification, semantic textual similarity, natural language inference, and question answering. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 3890–3902, Santa Fe, New Mexico, USA. 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. *CoRR*, abs/1907.11692.

Marius Mosbach, Maksym Andriushchenko, and Dietrich Klakow. 2021. On the stability of fine-tuning BERT: misconceptions, explanations, and strong baselines. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net.

Manikandan Ravikiran and Subbiah Annamalai. 2021. DOSA: Dravidian code-mixed offensive span identification dataset. In *Proceedings of the First Workshop on Speech and Language Technologies for Dravidian Languages*, pages 10–17, Kyiv. Association for Computational Linguistics.

Manikandan Ravikiran, Bharathi Raja Chakravarthi, Anand Kumar Madasamy, Sangeetha Sivanesan, Ratnavel Rajalakshmi, Sajeetha Thavareesan, Rahul Ponnusamy, and Shankar Mahadevan. 2022. Findings of the shared task on Offensive Span Identification in code-mixed Tamil-English comments. In *Proceedings of the Second Workshop on Speech and*

Language Technologies for Dravidian Languages. Association for Computational Linguistics.

Marco Túlio Ribeiro, Sameer Singh, and Carlos Guestrin. 2016a. "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.

Marco Túlio Ribeiro, Sameer Singh, and Carlos Guestrin. 2016b. "why should I trust you?": Explaining the predictions of any classifier. In *Proceedings of the Demonstrations Session, NAACL HLT 2016, The 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, San Diego California, USA, June 12-17, 2016*, pages 97–101. The Association for Computational Linguistics.

Peter Shaw, Jakob Uszkoreit, and Ashish Vaswani. 2018. Self-attention with relative position representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 464–468, New Orleans, Louisiana. Association for Computational Linguistics.

Sandipan Sikdar, Parantapa Bhattacharya, and Kieran Heese. 2021. Integrated directional gradients: Feature interaction attribution for neural NLP models. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 865–878, Online. 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, ICML 2017, Sydney, NSW, Australia, 6-11 August 2017*, volume 70 of *Proceedings of Machine Learning Research*, pages 3319–3328. PMLR.

James Thorne, Andreas Vlachos, Christos Christodoulopoulos, and Arpit Mittal. 2019. Generating token-level explanations for natural language inference. 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 963–969, Minneapolis, Minnesota. Association for Computational Linguistics.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, and Jamie Brew. 2019. Huggingface's transformers: State-of-the-art natural language processing. *CoRR*, abs/1910.03771.

Peng Xu, Dhruv Kumar, Wei Yang, Wenjie Zi, Keyi Tang, Chenyang Huang, Jackie Chi Kit Cheung, Simon J.D. Prince, and Yanshuai Cao. 2021. Optimizing deeper transformers on small datasets. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 2089–2102, Online. Association for Computational Linguistics.