A Study In Hate: Dissecting Transformer-Based Models' Rationale for Implicit Hate Classification

Faye Holt and **Cuong Nguyen** and **Parth Shah**
Georgia Institute of Technology

1 Abstract

Transformer-based models are widely used and very successful in a variety of language-based tasks. However, while their complicated structure allows for state-of-the-art results, it also poses a challenge in the space of explainability. Hate speech detection is one critical domain where explainability is crucial as hate speech becomes more prominent in online communities. In this paper, we interpret SVM and BERT models trained on an implicit hate speech dataset introduced by El Sherief et al. 2021 using state-of-the-art explainability frameworks for machine learning models, first and foremost being SHAP (Lundberg et al. 2017). After achieving comparable classification results on both SVM and BERT to previous models built on this dataset, we used SHAP to retrieve the top predictive features for both binary and multi-class classification. We find that the models seem put a heavy focus on gender and ethnic-based nouns, while ignoring hate verbs. In addition, we find that stopwords and punctuations are useful to models in distinguishing between hate and non-hate for certain sentences. Our work provides additional understanding to how Transformer-based models recognize hate speech, and how such faculty can be improved upon in the future.

2 Introduction and Related Work

Our research into explainability is motivated by a lack of understanding of transformer-based models in conjunction with the importance of implicit-hate classification. As transformers are a popular tool for hate classification, understanding their classifications is crucial in alleviating bias and ensuring that implemented models are placing weight on actual hate-speech. As such, the goal of our project is to figure out what words or patterns models are looking at when they are trained for the two tasks mentioned in ElSherief et al. (7):

- Binary classification between non-hate tweet (encoded as 0 in dataset) and hate tweets (implicit and explicit, encoded as 1 in dataset)
- Multi-class classification between fine-grained hate categories via the hate taxonomy introduced in ElSherief et al.

Over the last decade, transformer based models have steadily increased in popularity due to their wide use and success in a wide array of language based tasks. However, the techniques that transformer based models use are oftentimes difficult to understand. Bolukbasi et al. found that much of the difficulty in being able to interpret transformer based models resides in "interpretability illusion"; patterns that are interpreted could in fact be trivial. Our goal is to analyze different transformer based models that are trained on an implicit hate dataset. Our reason for choosing the implicit hate dataset is because we believe that detecting hate speech is becoming increasingly important as social media is reaching a larger audience.

Our trained models have a clear bias towards "identity words", which are words such as "Jew", or "White". This bias seems consistent with ElSherief et al.'s findings in their analysis of implicit versus explicit hate speech.

We also noticed that BERT seems to pay just as much attention to non-important tokens, such as punctuation and stop words, as nouns and verbs. We theorized that removing stop words and punctuation from speech would increase performance, but the opposite was true in our experiment. However, Moshkin et al. found that removing stop words in tweets actually improved the performance of their model. This issue could be specific to our problem area, which was implicit versus explicit hate, but more research needs to be done in this area to better understand the difference between the models.

Rogers et al. found that BERT embeddings

contain both syntactic and semantic information, which we find to be consistent with our analysis of self-attention. They also analyzed the specific function of the special tokens in self-attention, as it is not well understood as of now. We mention the same pattern in our findings, but have no definite solution as to why BERT seems to rely on these special tokens so much.

3 Methods

3.1 Data

As previously described, we use the implicit hate speech dataset that is described and provided by ElSherief et al. (7). The dataset contains 21480 tweets represented by their Tweet ID on the platform, and their respective labels under 2 different annotation schemes. We re-hydrated the tweets (i.e retrieve the body text) using the Twitter API. However, due to various Twitter restrictions we were only able to retrieve about 1/3 of the original dataset. Table 1 describes summary statistics of collected dataset given the binary classification scheme. Table 6 describe the number of documents collected per implicit hate category.

Label	Overall	Retriev.	Avg Length
Hate (1)	8189	2832	17.02
Non-Hate (0)	13291	5632	15.69
Total	21480	8464	16.14

Table 1: Summary Statistics for Dataset

Label	Retriev.	% Distribution
0 Stereotypical	433	0.21
1 White Grievance	391	0.19
2 Incitement	386	0.19
3 Threatening	182	0.09
4 other	14	0.01
5 Inferiority	226	0.11
6 Irony	400	0.2

Table 2: Implicit Hate Category Label Distribution

3.1.1 Data Preprocessing

As mentioned, the dataset contains a total of 8464 tweets after re-hydration using the Twitter API. The data that was not able to be retrieved was removed from the dataset. Each capital letter was replaced by the respective lowercase letter, and all URL's, non-ASCII characters, punctuation, and stop words were removed. For examples from the dataset please see Table 3 for binary classes and Table 4 for multi-class.

3.2 Models

To solve the problem of hate speech classification, we shall frame it as a sequence classification problem: Given a sequence of tokens $t_1, t_2, ...t_n$, we train a model that will assign to the a label l within the set of labels $l_1, l_2, ..., l_k$. To perform this task and subject for subsequent analysis, we consider the following models:

3.2.1 SVM

Support Vector Machine (3) is a discriminative classifier that aims to learn a hyperplane that best separates data points belonging to different classes. Kernels such as RBF or Polynomial (degree n) can be used to learn non-linear boundaries via the kernel trick. As one of the baselines, we trained an SVM model with $C = 1, kernel = rbf, scale = gamma$. We utilize SVM as a baseline model for comparison against transformers.

3.2.2 BERT

BERT is a revolutionary language model that was introduced in the paper "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding" (5) BERT uses Masked LM (MLM), which randomly masks words in the sentence and then tries to predict them. As another baseline to compare between Transformers-based models, we retrieved the pre-trained "bert-base-uncased" model from the Huggingface repository for fine-tuning (hidden-size = 3072, num-attention-heads = 12, num-hidden-layers = 12) using the Trainer function (learning-rate = 5e-5), train for 10 epochs while saving every 500 steps. For further analysis, we select the model checkpoint with the highest validation F1

3.2.3 ELECTRA and DistilBERT

ELECTRA is an Transformer-based model introduced in "Pre-training Text Encoders As Discriminators Rather Than Generators" (4) Instead of corrupting the input by replacing tokens with "[MASK]" as in BERT, ELECTRA corrupts the input by replacing some input tokens with incorrect, but somewhat plausible, fakes. We retrieved the pre-trained "google/electra-small-discriminator"

Label	Tweet Example
Not Hate	RT @ANInews: Ahmad Javed appointed as the next Ambassador of India to the Kingdom of Saudi Arabia.
Hate	2 Hollywood Jews want us to feel bad for not importing jihadists whose doctrine has more Jew hate than Mein Kampf https://t.co/15cP3LgCHo

Table 3: Binary Classification Examples

Label	Tweet Example
Stereotypical	RT @jihadwatchRS: UK: Students in Islamic schools taught Sharia amputation methods and that Jews are conspiring to take over... http://t ...
White Grievance	@ANP14 And of course, "anti-racist" is a code word for "anti-white". "Self-hating Jews" are nothing compared with self-hating white people.
Incitement	Ukraine Ukrainian Thanks brothers. White Pride Worldwide. wpww 14words. @WhiteResister https://t.co/nzlxoMqmkn
Threatening	@FifthKentucky @occdissent They Smother the Good White People of America! They Need to be Hung, Brother "South"!
Inferiority	@CheetoCrumbles @jbrownsghost @occdissent You hate white people because we're better looking and we're smarter.
Irony	@DerekC2016 @FAIRImmigration @lshi7175 @BreitbartNews @krod315 Yes jail them so they can cost taxpayers more $ than their food stamps did
Other	"@AmRenaissance: Jared Taylor on the white man's disease: pathological altruism. http://t.co/mzBBEQi0 race tpot tcot"....WOW! a must read

Table 4: Implicit Hate Multi-Class Examples

model from the Huggingface repository for fine-tuning (hidden-size = 256, num-attention-heads = 4, num-hidden-layers = 12) with the same training regime as above. DistilBERT is another Transformer-based model introduced in "DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter". We retrieved the pre-trained "distilbert-base-uncased" model from the Huggingface repository for fine-tuning (hidden-size = 3072), num-attention-heads = 6, num-hidden-layers = 12) with the same training regime as above.

3.2.4 Model Results

Label	SVM (baseline)	BERT
Precision	0.49	0.656
Recall	0.68	0.71
F1	0.59	0.693
Accuracy	0.76	0.77
	ELECTRA	**DistilBERT**
Precision	**0.737**	0.684
Recall	**0.706**	0.677
F1	**0.721**	0.68
Accuracy	**0.812**	0.794

Table 5: Binary Model Results (Macro-Averaged)

Label	SVM (bl) Acc	BERT Acc
0 Stereotypical	0.62	**0.73**
1 White Grievance	**0.63**	0.14
2 Incitement	**0.5**	0.18
3 Threatening	**0.38**	0.25
4 Other	0	0
5 Inferiority	**0.14**	0
6 Irony	**0.74**	0.68
F1	**0.43**	0.28

Table 6: Multi-Class Model Results (Accuracy and F1)

Our F1 scores and success metrics for these models are not ideal as we were not able to retrieve all tweets in the dataset, however, since our project's focus is on explainability and not model construction, we plan to continue with the current model metrics.

3.3 Explainability

3.3.1 SHAP

We use SHAP to examine how certain words within a sentence may influence the model's final classification decisions. Our decision to use SHAP rather than other explainability methods stems from the fact that SHAP is not only model-agnostic, but also

the most theoretically-sound explainability framework out of the available options. This is due that fact that SHAP feature scores can not only be calculated for localized samples, but also for the entire global dataset. SHAP is based on top of SHAP values, a game-theoretic concept that intuitively describes each feature's contribution to the final outcome, after taking into account all possible combinations of features. Given the additive feature attribution method for an explanation model $g(\cdot)$ with the form:

$$g(z') = \phi_0 + \sum_{j=1}^{n} \phi_j * z'_j$$

where n is the number of simplified features, and z'_j is an indicator variable for whether the simplified input appears in z'. SHAP hypothesized that with the correct weighting setup, ϕ_j will be the Shapley value for the given feature j. The *shap* library written by the authors provides various different methods to approximate the Shapley values based on the model type, including kernelSHAP (used for SVM), and deepSHAP (used for BERT).

3.3.2 BERT Visualizer

We use BERT visualizer to visualize attention for our BERT model. Compared to other visualization tools, BERT visualizer provides many different views that can help to visualize the attention patterns in different ways. It can provide a view of every attention head at once, an aggregate of a single layer with multiple heads at a time, or even a visualization of query and key vectors, as well as the product.

4 Results

4.1 Feature Difference Across Models

We are interested in gaining a better understanding of what features each of our explored models emphasizes. We are particularly interested in the difference between Binary and Multi-Class BERT and exploring how feature manipulation effects which features are given the most weight.

4.1.1 Experiment Setup

We explore the difference in important features across models using SHAP Values. For Binary and Multi-Class BERT, we generate SHAP values on a set of randomly sampled sentences from our dataset and output the 15 most weighted features across categories (Table 7). We then mask the top 3 features outputted in these SHAP Values and generate new SHAP Values on our masked data to see how our features change. After noting the common pattern of protected class nouns appearing in both models, we completed another round of feature masking, this time masking protected classes. The bolded words in Table 7 are the features we masked. Figure 1 displays the results of this masking on one sample sentencing from the *Threatening* class.

4.1.2 Results Comparison

When comparing across Binary versus Multi-Class BERT, we saw many similarities in feature importance. Of particular note is the common pattern of the most important features being protected class nouns defined by Equal Opportunity Law as "Groups includ[ing] men and women on the basis of sex; any group which shares a common race, religion, color, or national origin; people over 40; and people with physical or mental handicaps." (6). Examples of some of these nouns are bolded in Table 7. When we masked these particular features, we saw a shift in feature importance to more explicit hate language such as, "hunt", "racist", and "kill", see Figure 1 for an example. This finding reveals that both Binary and Multi-Class Transformers are learning to recognize the targets of implicit hate (which is often protected classes) for classification, rather than the language that is attacking them. However, when these targets are masked, the models are able to effectively pick up on patterns of hate-speech, indicating that future work may explore training the models with masked protected classes.

Contrasting between binary and multi-class models, we did not see the results we initially hypothesized of binary and multi-class models having significant differences. While there are differences in feature importance, there is not a clear pattern present from qualitative feature evaluation. Therefore, to gain a better understanding of this aspect of our research, we next explore attention in binary and multi-class BERT models.

4.2 Attention Patterns in BERT

To better understand how transformer architecture based models like BERT are trained, we want to explore how these models can use their attention mechanisms to pick up the syntactic meaning of the input. More specifically, we are interested in what kind of attention patterns BERT uses to assign importance weights.

BERT Binary	BERT Multi-class					
Hate	Irony	Stereotypical	White Grievance	Incitement	Threatening	Inferiority
wake	getting	wants	–	–	–	wants
whites	tile	getting	wants	getting	brotherhood	tman
rape	**islam**	**ukraine**	change	violent	violent	utation
sell	gen	com	etc	**ukraine**	com	call
je	ev	gen	come	–	person	p
islamic	etc	per	crime	com	pen	thirty-one
try	booklet	crime	com	holm	forty-seven	ong
amnesty	wants	hurting	groups	wants	tile	tion
white	**ukraine**	make	murders	forty-seven	nazis	come
jewish	come	thirty-one	solidarity	support	holm	holm
black	nazis	festival	tman	shoved	words	**ukraine**
council	festival	attending	network	per	american	king
kill	**quran**	come	words	sign	**jews**	screwed
alien	**jewish**	**islam**	gen	crime	biggest	tics

Table 7: Top 15 weighted features for binary and multi-class models. Highlights indicate similar features across binary and multi-class. Bolded words indicate features chosen to mask.

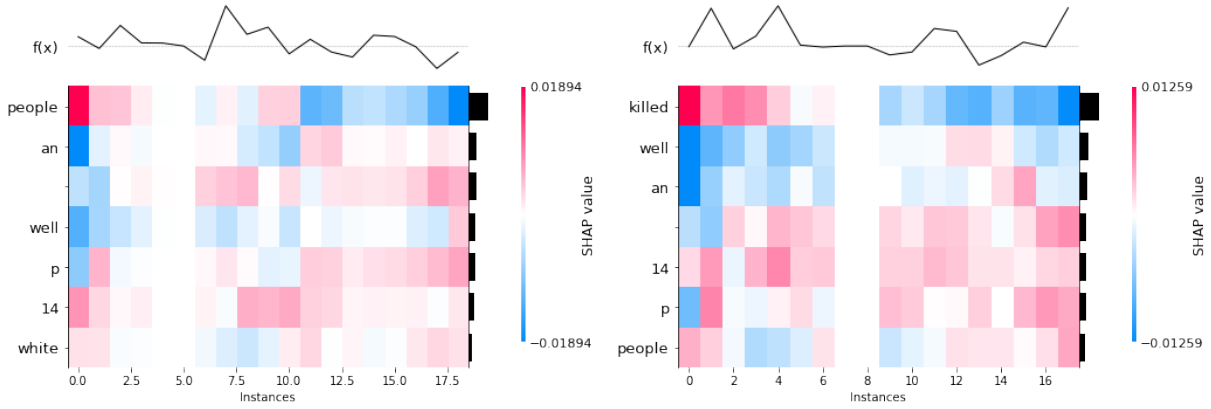

Figure 1: Most important features in BERT Multi-Class for the sample sentence: *"well white people killed lot people maybe good thing maybe kill vile nazis like" Label: Threatening*. Left figure is pre-feature masking and right figure is post-feature masking.

4.2.1 Experiment Setup

Using our (standard 12 layer, 12 head) BERT model and a BERT visualization tool called BertViz, we can visualize the attention given an input. We then take a set of sentences, both implicit and explicit hate, from our dataset and display attention using the visualizer. We can then visually analyze the attention patterns of each input over individual layers and heads. Common patterns will appear consistently over the inputs (Figure 3).

4.2.2 Result Comparison

Over all inputs, we consistently saw a few patterns emerge over the layers. In particular, the main patterns that emerged were attending to delimiters (like the SEP token), attending to the word placed immediately before or after the word in the sentence, and a distributed attention weighted almost evenly over the sentence. Interestingly, a significant portion of attention is focused on the special tokens like SEP and CLS, and earlier layers tend to focus on the CLS token while a heavier weight on the SEP token is more common in the later layers. Despite a vast majority of the attention heads being

Removal Strategy	F1
Base (No Removal)	**0.64**
Punctuations	0.6240
Stopwords	0.6229
Twitter Artefacts (TA)	0.6156
Punctuations + Stopwords + Twitter Artefacts	0.5911

Table 8: Results for Ablation Test

made up some combination of the above patterns, there are a few heads that focus on, what could be assumed to be, words that are related grammatically. For example, assigning a heavier weight from a direct object to a corresponding verb, between related nouns, or a possessive relationship. Evidence of these kinds of patterns suggests that BERT's attention heavily weighs tokens with their direct context as well as having special tokens be representations of the entire input altogether.

4.3 Feature Ablation on Stop Elements

4.3.1 Experiment Setup

We want to test whether or not seemingly non-informative elements within sentences such as punctuations and stopwords have any influence on our model's ability to distinguish between hate and non-hate sentences. To do this, we test the performance of our BERT model (described above), trained and tested using the same train-test split as previously mentioned, but now will be subjected to the following strategies:

- No Removal (Control)
- Remove all punctuations, as defined by Python's string.punctuations list
- Remove all stopwords, as defined by NLTK's English stopwords list
- Remove all Twitter mentions, hashtags, and URLs using the tweet-preprocessor library
- Perform all three of the above removal strategies

4.3.2 Results Comparison

Table 8 shows the results for our ablation test. We notice that every removal strategy leads to a decrease in performance, with a combination of all three strategies leading to a sharpest decrease in performance (0.05 F1). We also notice that removing Twitter Artefacts lead to the largest individual decrease in performance (0.01 more than the other two). This could be due to the fact that the model is learning to distinguish hashtags and mentions that are popular in hate instances

Punctuation	BERT	ELECTRA	DistilBERT
!	0.4006	2.149	1.4377
"	0.2647	1.713	0.4288
#	0.2413	0.6521	0.3909
’	2.1152	2.6717	0.8901
(	0.0883	0.4376	0.0548
)	0.0197	0.0503	0.0399
,	0.7749	0.6413	0.7259
.	1.2342	2.1705	1.6963
/	0.3822	1.32	0.1336
?	0.9752	1.8565	3.6899
@	0.531	1.8426	0.6023

Table 9: Maximal SHAP Values for selected punctuations (with logit scaling)

To quantitatively measure the importance of these elements to our models in classifying hate vs non-hate, we record their maximum SHAP values after running the Explainer on the same SHAP sentence sample as described above. Table 9 shows the aforementioned results for our three Transformer-based models: BERT, ELECTRA, and DistilBERT We notice that even though punctuations have a relative low average SHAP values, some punctuation marks such as "?" or "’" have relatively high maximum SHAP values (with "?" even being in the top 10 for the case of DistilBERT). One explanation for this phenomena is the fact that many hate samples in our dataset utilized punctuations to aid their hateful message. Figure 3 shows an hate sample where the question mark plays a crucial role in facilitating the question and answer format of the offensive joke.

4.4 Work Division

Our work division for the final report was to split up answering the research questions. Faye Holt worked on research question one regarding feature difference across models, Parth Shah worked on research question two, attention patterns in BERT, and Johnny Nguyen worked on research question three about feature ablation on stop elements, in addition to contributing towards RQ1 for the binary case. Our work division up until the midpoint

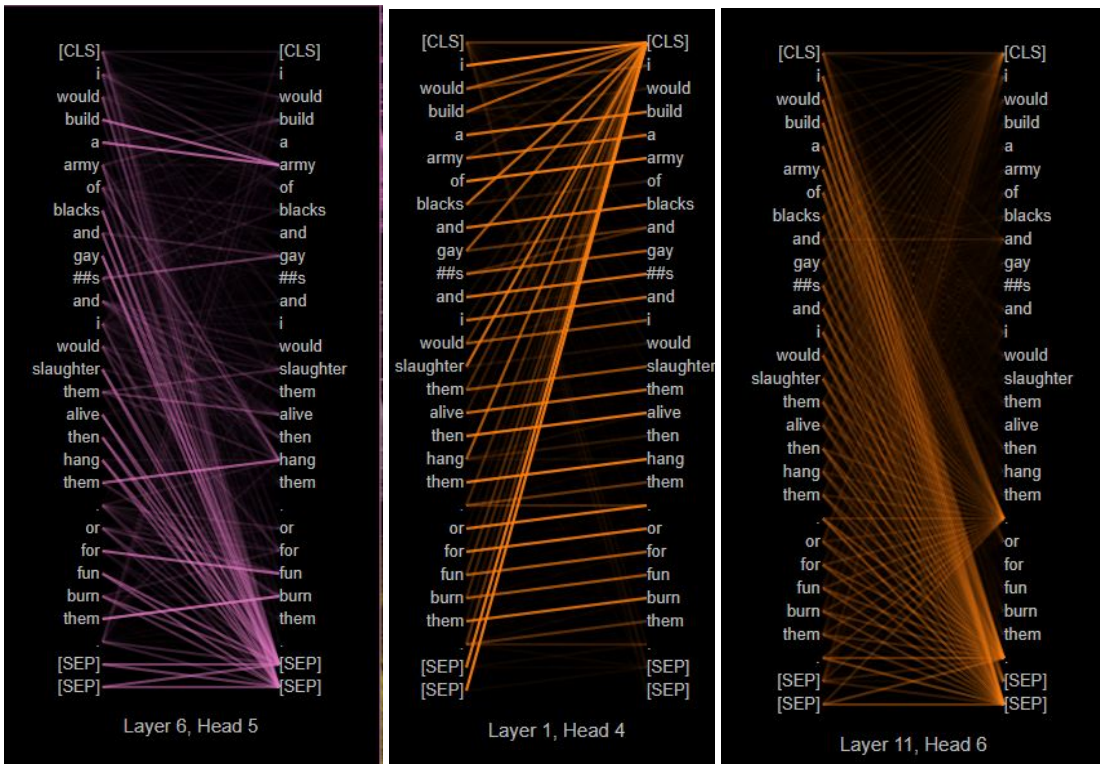

Figure 2: Common attention patterns

focused on dividing analysis based on models to ensure that everyone is involved/ understanding each step in the analysis. Parth Shah trained the SVM binary and multi-class models. Cuong Nguyen did the same for BERT Binary and Faye Holt for BERT multi-class. Each team member calculated SHAP values and generated plots for their respective models.

5 Conclusion

When comparing models, we found many similarities among feature importance. The most important feature were identity based nouns which is consistent with ElShereif et al.'s findings. However, when these features were masked, explicitly negative language was more prominent according to the SHAP values. This shows that our models are learning that sentences with identity based nouns are the main indicator of hate speech. In the future, more research could be done to train models while masking protected classes, so that the model could attempt to focus on a more varied approach to hate speech identification.

Through our experiment, we also found that removing stop words or punctuations reduced the performance of our models. We hypothesize that this is due to stop words and punctuation containing syntactic meaning, so removal of syntactic meaning in the sentence would hinder our model. This seems inconsistent with other findings, which report that removing stop words and punctuation offers very little change in the performance of the model (Moshkin et al). For future work, we want to identify why our findings are inconsistent with other works, and if this problem is specific to our model, dataset, or problem area.

Lastly, through analyzing attention patterns in BERT, we have gained a much better understanding of how the transformer model assigns weights throughout each layer and head. There is heavy importance on word distance, as well as identification of "linked" words such as possessive's and direct objects. We also found that special tokens are considered very important to the attention model. We hypothesize that this leads to special tokens containing sentence-level syntactic and semantic meanings, although this is something that needs to be further researched.

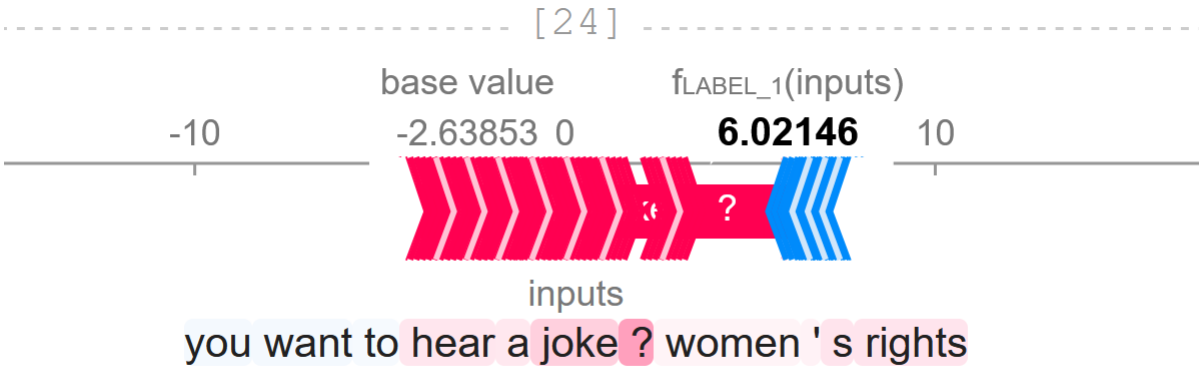

Figure 3: Hateful Instance where the question mark played a crucial role in driving the model's correct classification

6 Ethical, Broader Impacts Statement

This research into the explainability of transformer-based models was based in the context of implicit-hate classification. Our initial motivation behind this research was the important social implications of correctly classifying hate in an era of social media dominance. As these platforms continue to grow, the monitoring of hate increases in importance. Yet, of equal, if not greater importance, is classification of hate that does not discriminate against any one language, dialect, or group. Previous research has found there is an alarming degree of racial bias in automatic hate classification tools, particularly on platforms like Twitter incorrectly flagging AAVE speech as hate speech (1). The present bias of hate-speech classification creates an ironic ethical dilemma wherein models designed to alleviate discrimination end up being a source or prejudice.

Our research, similarly, found potential patterns of bias in the transformer-based implicit hate classification models. As stated in section 4.1, the most prominent features across models is protected noun classes. This findings indicated transformer-based models may incorrectly identify implicit hate if a typical target of hate is mentioned in an utterance. Although research on dialectal bias is currently prominent, we found less research focused on this feature explainability in relation to bias. Therefore, we propose that broader impacts of our research could be further exploration into feature bias in transformer-based models, as well as research that explores how to mask protected classes when training implicit hate models.

7 Code Repository

4650-Implicit-Hate-Study Github

References

[1] Ahmed, Zo. Vidgen, Bertie. Hale, Scott A. . Tackling Racial Bias in Automated Online Hate Detection: Towards Fair and Accurate Classification of Hateful Online Users Using Geometric Deep Learning. *arXiv preprint arXiv*:2103.11806

[2] Bolukbasi, T., Pearce, A., Yuan, A., Coenen, A., Reif, E., Viégas, F., Wattenberg, M. (2021). An Interpretability Illusion for BERT. *arXiv preprint arXiv*:2104.07143.

[3] Cortes, C., Vapnik, V. (1995). Support-vector networks. *Machine learning*, 20(3), 273-297.

[4] Clark, K., Luong, M. T., Le, Q. V., Manning, C. D. (2020). Electra: Pre-training text encoders as discriminators rather than generators. *arXiv preprint arXiv*:2003.10555.

[5] Devlin, J., Chang, M. W., Lee, K., Toutanova, K. (2018). Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv*:1810.04805.

[6] Equal Employment Opportunity National Archives. https://www.archives.gov/eeo/terminology.html, Accessed: 2021-12-10.

[7] ElSherief, M., Ziems, C., Muchlinski, D., Anupindi, V., Seybolt, J., De Choudhury, M., Yang, D. (2021). Latent Hatred: A Benchmark for Understanding Implicit Hate Speech. *arXiv preprint arXiv*:2109.05322.

[8] Goldstein, A., Kapelner, A., Bleich, J., Pitkin, E. (2015). Peeking inside the black box: Visualizing statistical learning with plots of individual conditional expectation. journal of Computational and Graphical Statistics, 24(1), 44-65.

[9] Lundberg, S. M., Lee, S. I. (2017, December). A unified approach to interpreting model predictions. *In Proceedings of the 31st international conference on neural information processing systems* (pp. 4768-4777).

[10] Moshkin V., Konstantinov A., Yarushkina N. (2020) Application of the BERT Language Model for Sentiment Analysis of Social Network Posts. In: Kuznetsov S.O., Panov A.I., Yakovlev K.S. (eds) Artificial Intelligence. RCAI 2020. Lecture Notes in Computer Science, vol 12412. Springer, Cham. *https://doi.org/10.1007/978-3-030-59535-7*$_2$0

[11] Nori, H., Jenkins, S., Koch, P., Caruana, R. (2019). Interpretml: A unified framework for machine learning interpretability. *arXiv preprint arXiv*:1909.09223.

[12] Ribeiro, M. T., Singh, S., Guestrin, C. (2016, August). " 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).

[13] Rogers, A., Kovaleva, O., Rumshisky, A. (2020). A primer in bertology: What we know about how bert works. Transactions of the Association for Computational Linguistics, 8, 842-866.

[14] Rychener, Y., Renard, X., Seddah, D., Frossard, P., Detyniecki, M. (2020). Sentence-Based Model Agnostic NLP Interpretability. *arXiv preprint arXiv*:2012.13189.

[15] Xiang, T., MacAvaney, S., Yang, E., Goharian, N. (2021). ToxCCIn: Toxic Content Classification with Interpretability. *arXiv preprint arXiv*:2103.01328.

[16] Sanh, V., Debut, L., Chaumond, J., & Wolf, T. (2019). DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter. arXiv preprint arXiv:1910.01108.