

# HATEMOJI: A Test Suite and Adversarially-Generated Dataset for Benchmarking and Detecting Emoji-Based Hate

Hannah Rose Kirk<sup>1,2‡</sup>, Bertie Vidgen<sup>1,2</sup>, Paul Röttger<sup>1,2</sup>, Tristan Thrush<sup>3\*</sup>, Scott A. Hale<sup>1,2,4</sup>

<sup>1</sup>University of Oxford, <sup>2</sup>The Alan Turing Institute, <sup>3</sup>Hugging Face, <sup>4</sup>Meedan

<sup>‡</sup>hannah.kirk@oii.ox.ac.uk

## Abstract

Detecting online hate is a complex task, and low-performing models have harmful consequences when used for sensitive applications such as content moderation. Emoji-based hate is an emerging challenge for automated detection. We present HATEMOJICHECK, a test suite of 3,930 short-form statements that allows us to evaluate performance on hateful language expressed with emoji. Using the test suite, we expose weaknesses in existing hate detection models. To address these weaknesses, we create the HATEMOJIBUILD dataset using a human-and-model-in-the-loop approach. Models built with these 5,912 adversarial examples perform substantially better at detecting emoji-based hate, while retaining strong performance on text-only hate. Both HATEMOJICHECK and HATEMOJIBUILD are made publicly available.<sup>1</sup>

**Content Warning** This article contains examples of hateful language from HATEMOJICHECK to illustrate its composition. Examples are quoted verbatim, except for slurs and profanity in text, for which the first vowel is replaced with an asterisk. The authors oppose the use of hateful language.

## 1 Introduction

Online hate harms its targets, disrupts online communities, pollutes civic discourse and reinforces social power imbalances (Gelber and McNamara, 2016). The sheer scale of hateful content online has led to widespread use of automated detection systems to find, monitor and stop it (Vidgen et al., 2019; Gillespie, 2020). However, hateful content is complex and diverse, which makes it challenging to detection systems. One particular challenge is the use of emoji for expressing hate. Emoji are pictorial representations that can be embedded in text,

allowing complex emotions, actions and intentions to be displayed concisely (Rodrigues et al., 2018). Over 95% of internet users have used an emoji and 10 million are sent every day (Brandwatch, 2018). Following England’s defeat in the Euro 2020 football final, there was widespread racist use of emoji such as 🙄, 🍌 and 🍷 (Jamieson, 2020). This paper focuses on emoji-based hate, answering two research questions:

**RQ1** What are the weaknesses of current detection systems for hate expressed with emoji?

**RQ2** To what extent can human-and-model-in-the-loop training improve the performance of detection systems for emoji-based hate?

To answer RQ1, we present HATEMOJICHECK, a suite of functional tests for emoji-based hate. We provide 3,930 test cases for seven functionalities, covering six identities. 2,126 original test cases are matched with three types of challenging perturbations to enable accurate evaluation of model decision boundaries (Gardner et al., 2020). We use HATEMOJICHECK to assess Google Jigsaw’s Perspective API as well as models trained on academic datasets, exposing critical model weaknesses.

To answer RQ2 and address the model weaknesses identified by HATEMOJICHECK, we implement a human-and-model-in-the-loop dynamic training scheme. We build on the work of Vidgen et al. (2021b), who used this approach for textual hate. Our work begins where their study ends. We conduct three rounds of adversarial data generation focused explicitly on emoji-based hate, tasking annotators to generate sentences that the model-in-the-loop misclassifies. This process yields a dataset of 5,912 entries, half of which are challenging contrasts. We call this dataset HATEMOJIBUILD. The dataset is evenly split between hate and non-hate, and each hateful entry has labels for the type and target. Between each round, the model-in-the-loop

\* TT’s work was done at Facebook AI Research.

<sup>1</sup>See our [Github Repository](#). HATEMOJICHECK, HATEMOJIBUILD and the final HATEMOJI Model are also available on HuggingFace.

is re-trained so that annotators are trying to trick a progressively stronger and more ‘emoji-aware’ model. Relative to existing commercial and academic models, our models improve performance on the detection of emoji-based hate, without sacrificing performance on text-only hate.

We make several contributions: (1) we construct HATEMOJICHECK, which tests key types of emoji-based hate as separate functionalities, (2) we evaluate the performance of existing academic and commercial models at detecting emoji-based hate, (3) we present HATEMOJIBUILD, a labeled emoji-based hate speech dataset that is adversarially-generated for model training and (4) we train models that can accurately detect emoji-based hate. These data-centric contributions demonstrate the benefits of systematic and granular evaluation, and the need to diversify how hate detection systems are trained. We make both HATEMOJICHECK and HATEMOJIBUILD publicly available.

**Definition of Hate** We use the United Nations definition of hate speech: “any kind of communication in speech, writing or behavior, that attacks or uses pejorative or discriminatory language with reference to a person or a group on the basis of who they are, in other words, based on their religion, ethnicity, nationality, race, color, descent, gender or other identity factor” (United Nations, 2019, p.2).<sup>2</sup>

## 2 HATEMOJICHECK: Functional Tests for Emoji-Based Hate

The concept of functional testing, an expected input-output behavior (Beizer, 1995), has been adapted from software engineering for NLP tasks (Ribeiro et al., 2020), including hate speech detection (Röttger et al., 2021). These diagnostic tests identify model vulnerabilities to intentionally simple and non-ambiguous constructions of hate so comprise a minimal performance standard.

### 2.1 Identifying Functionalities

A functionality describes the ability of a model to provide a correct classification when presented with hateful or non-hateful content (Röttger et al., 2021). Each functionality has a set of corresponding test cases that share one gold-standard label. We select the functionalities in HATEMOJICHECK to be (1) realistic: they capture real-world uses of

<sup>2</sup>We recognize the presence of annotator bias in cultural interpretations of this definition, and provide data statements for both datasets in the appendix.

emoji-based hate, (2) unique: each covers a distinct aspect of emoji-based hate, without overlaps between functionalities, and (3) unambiguous: they have clear gold-standard labels. The functionalities are motivated from two perspectives:

**Previous Literature** We identify distinct uses of emoji in online communications, particularly abuse. This includes appending emoji to the end of otherwise neutral statements to reveal emotive intent (Wiegand and Ruppenhofer, 2021), using emoji as a ‘pictorial slur’ to attack or dehumanize groups (Bick, 2020; Wiegand and Ruppenhofer, 2021) and substituting emoji for identity terms or threatening verbs (Samghabadi et al., 2019; Bick, 2020; Wiegand and Ruppenhofer, 2021). We also review a range of literature on online hate, identifying distinct types of hateful language (Talat et al., 2017; Zampieri et al., 2019; Vidgen et al., 2021a).

**Empirical Investigation** Based on the literature, we define a list of potentially hateful emoji and words, and use Twitter’s Streaming API to search for the Cartesian products of emoji–emoji and emoji–word pairs over a two week period. To identify different forms of emoji-based hate, we apply a grounded theory approach (Corbin and Strauss, 1990) on a sample of 3,295 tweets, splitting out distinctive categories, and recursively selecting sub-categories until all key parts of the data are captured and the framework is ‘saturated’.

### 2.2 Functionalities in HATEMOJICHECK

This section presents verbatim examples of emoji-based hate (in gray text). For masked versions, see Tab. 1 and Tab. 4. HATEMOJICHECK has seven functionalities (see Tab. 1). **F1 Verb Swap** tests threatening hate where the verb is replaced with the equivalent emoji: 🗡️, 🍷, 📢, 🗡️, 🗡️, 🗡️. It includes both direct threats and normative statements of threat. **F2 Identity Swap** tests derogatory hate where the identity term is replaced with an emoji representation: woman (👩), trans people (👩🏳️‍🌈), gay people (🏳️‍🌈), disabled people (♿), Black people (👩🏿) and Muslims (👤).<sup>3</sup> **F3 Descriptor Swap** tests hate where the negative descriptor is replaced with the equivalent emoji. It includes both dehumanizing language and expressions of disgust. For dehumanizing language, five descriptors are used: 🐼, 🐼, 🐼, 🐼, 🐼. Based on our Twitter analysis, target-specific descriptors are

<sup>3</sup>👤 was commonly used to denote Muslims on Twitter, despite its Unicode description being ‘person wearing a turban’.

Table 1: Summary of the seven functionalities in HATEMOJICHECK.

Functionality (Emoji Used)	Templates	Emoji	Identities	Cases
<i>F1 Verb Swap</i> : Swap threatening verb (🔪, 🌪️, 🗡️, 🏹, ☠️)	10	5	6	300
<i>F2 Identity Swap</i> : Swap protected identity (👤, 🏳️‍🌈, 🇸🇰, 🇬🇧, 🇮🇳)	20	1	6	120
<i>F3 Descriptor Swap</i> : Swap descriptor noun/emotion (👎, 🤢, 🤮, 🤨, 🤬, 🤯, 🤪, 🤫, 🤭, 🤥, 🤦, 🤧, 🤩, 🤭, 🤮, 🤯, 🤪, 🤫, 🤭, 🤥, 🤦, 🤧)	10	3–7	6	260
<i>F4 Double Swap</i> : Combine two of the above swaps	10	3–7	6	288
<i>F5 Append</i> : Append negative emoji to neutral text (👎, 🤢, 🤮, 🤨, 🤬, 🤯, 🤪, 🤫, 🤭, 🤥, 🤦, 🤧)	10	3–5	6	288
<i>F6 Positive Confounder</i> : Append positive emoji to hateful text (❤️, 😊, 😄, 😍, 😂, 🥰, 🥳, 🥹, 🥰, 🥳, 🥹)	13	5	6	440
<i>F7 Emoji Leetspeak</i> : Substitute for character or word-piece (0, 1, 4, 3, *, 🇸🇰, 🇬🇧, 🇮🇳)	15	3–4	6 (+16 slurs)	430
<b>Totals of Unique Templates, Emoji, Identity and Test Cases</b>	<b>88</b>	<b>36</b>	<b>6</b>	<b>2,126</b>

used for Black people: (👮, 🏠) and Muslims: (👮, 🏠). For expressions of disgust, three descriptors are used: 🤢, 🤮, 🤨. **F4 Double Swap** tests a combination of F1 with either F2 or F3, where emoji are swapped for both the identity and the verb, or both the identity and the negative descriptor. Some templates contain only emoji e.g., “🔪👤”, while others have some text e.g., “👎 make me 🤢”. **F5 Append** tests statements which would be non-hateful on their own but become hateful when appended with a negative emoji such as 🤢, 🤮, 🤨. For instance, “My new manager is a woman” is neutral but “My new manager is a woman 🤢” is hateful. **F6 Positive Confounder** tests clearly hateful statements appended with a positive emoji such as 😊, 😄, 🥰. For instance, “Kill all women 😊” or “All women deserve to die 🥰”. **F7 Emoji Leetspeak** tests statements where parts of the hateful language have been obfuscated by replacing characters with emoji. The character substitution can occur in the term expressing negative sentiment, e.g., “I h4te gays”, the identity, e.g., “I hate g4ys” or the slur, e.g., “I hate f4gs”. There are three emoji character substitutions: (1) a numeric substitution which mimics leetspeak (a:4, i:1, e:3, o:0), (2) a star mimicking \* for vowels in slurs and profanities (vowel:★), and (3) a heart mimicking starring out vowels but in a more adversarial manner (vowel:♥). To reflect specific cases on Twitter of anti-Muslim hate, two additional slurs are included where a word-piece is substituted with an emoji (👮-f\*cker, 🏠-f\*cker).

## 2.3 Test Cases in HATEMOJICHECK

HATEMOJICHECK is strongly inspired by empirical examples of emoji-based hate but we want it to comprise a clear, minimum performance standard. Thus, for each functionality we hand-craft test cases, i.e., short English-language text documents that clearly correspond to just one gold-standard label. The test cases are varied in their vocabulary and syntax to ensure they assess model robustness

to diverse language (Zhou et al., 2020). To generate test cases at scale and ensure balanced coverage across targeted groups, we use a template-based approach (Dixon et al., 2018; Ribeiro et al., 2020; Röttger et al., 2021). First, each template is given an identity placeholder e.g., “I hate [IDENTITY]” or a slur placeholder e.g., “You are just a [SLUR] to me”. Six protected identities populate these placeholders: women, trans people, gay people, Black people, disabled people and Muslims. For templates with slurs, we use the 16 slurs compiled by Röttger et al. (2021) (see Appendix C.1). Second, each template is given an emoji placeholder that is populated by emoji that share semantic and syntactic properties, so that the test suite can be scaled across different emoji. For example, the template “I will [EMOJI] all [IDENTITY]” could be populated with {👎, ‘trans people’}. HATEMOJICHECK comprises 3,930 entries. There are 2,126 original test cases and 1,804 perturbations (see Tab. 1).

### 2.3.1 Perturbations

Each set of test cases is matched to a set of perturbations which are constructed through minimal edits to the original hateful template.<sup>4</sup> Test cases generated from these perturbed templates share vocabulary and syntax with the original test cases, enabling fine-grained assessment of model weaknesses (Gardner et al., 2020). We provide three types of perturbations. For **identity perturbations**, the identity targeted by the original hateful statement is substituted for a non-protected entity, making it non-hateful. Possible identity perturbations include non-protected groups (e.g., “I hate accountants”), non-human entities (e.g., “I hate spiders”), and inanimate objects (e.g., “I hate pizza”). For **polarity perturbations**, the negative sentiment of the original hateful statement is reversed to be positive, making it non-hateful, without changing the target of the statement. Possible polarity per-

<sup>4</sup>See Appendix C.2 for details on perturbations.

turbations include changing threatening language to supportive language, e.g., “kill” to “respect”, changing expressions of hate to expressions of love, e.g., “I hate...” to “I love...”, or replacing hateful use of slurs with counterspeech against the slurs, e.g., “[IDENTITY] are [SLUR]” to “[IDENTITY] should never be called [SLUR]”. For **no emoji perturbations**, emoji are removed or replaced with their equivalent text to preserve the semantic expression (e.g., “🔫” becomes “shoot” and “❤️” becomes “love”). For most functionalities this perturbation preserves the original label of the test case, e.g., “[IDENTITY] makes me 🤢” is hateful, and its perturbation “[IDENTITY] makes me sick” is still hateful. However, for F5 Append the label changes because when the negative emoji appended to the neutral statement is removed, the part of the statement that remains is non-hateful.

## 2.4 Validating Test Cases

To validate the gold-standard labels assigned to each test case, we recruited three annotators with prior experience on hate speech projects.<sup>5</sup> Annotators were given extensive and prescriptive guidelines (Röttger et al., 2022), as well as test tasks and training sessions, which included examining real-world examples of emoji-based hate from Twitter. We followed guidance for protecting annotator well-being (Vidgen et al., 2019). There were two iterative rounds of annotation. In the first round, each annotator labeled all 3,930 test cases as hateful or non-hateful, and had the option to flag unrealistic entries. Test cases with any disagreement or unrealistic flags were reviewed by the study authors ( $n = 289$ ). One-on-one interviews were conducted with annotators to identify dataset issues versus annotator error. From 289 test cases, 119 were identified as ambiguous or unrealistic, replaced with alternatives and re-issued to annotators for labeling. No further issues were raised. We measured inter-annotator agreement using Randolph’s Kappa (Randolph, 2005), obtaining a value of 0.85 for the final set of test cases, which indicates “almost perfect agreement” (Landis and Koch, 1977).

## 3 Building Better Models with HATEMOJIBUILD

As reported in §4, we find existing models perform poorly on emoji-based hate as measured with HATEMOJICHECK. We address those failings by

<sup>5</sup>See annotator demographics in Appendix B.

implementing a human-and-model-in-the-loop approach using the Dynabench interface in order to train a model that better detects emoji-based hate.<sup>6</sup>

**Dataset Generation** We implemented three successive rounds of data generation and model re-training to create the HATEMOJIBUILD dataset. In each round we tasked a team of 10 trained annotators with entering content that the model-in-the-loop would misclassify.<sup>7</sup> We refer to this model as the *target model*. Annotators were instructed to generate linguistically diverse entries while ensuring each entry was (1) realistic, (2) clearly hateful or non-hateful and (3) contained at least one emoji. Each entry was first given a binary label of hateful or non-hateful, and hateful content was assigned secondary labels for the type and target of hate (Zampieri et al., 2019; Vidgen et al., 2021a,b). Each entry was validated by two additional annotators, and an expert resolved disagreements. After validation, annotators created a perturbation for each entry that flipped the label. To maximize similarity between originals and perturbations, annotators could either make an emoji substitution while fixing the text or fix the emoji and minimally change the surrounding text. Including a hateful and non-hateful version of each sentence in the dataset prevents overfitting to certain emoji or identity references. Each perturbation received two additional label annotations, and disagreements were resolved by the expert. This weekly cadence of annotator tasks was repeated in three consecutive weeks. The dataset composition and inter-annotator agreement is described in Appendix E.

**Model Implementation** Our work follows directly from Vidgen et al. (2021b), who collect four rounds (R1–4) of dynamically-generated textual data as well as 468,928 entries compiled from 11 English-language hate speech datasets (R0). Our rounds of data collection are referred to as R5–7. At the end of each round, the data for that round is assigned a 80/10/10 train/dev/test split. The test split is never used for any future training. The train split is upsampled to improve performance with multipliers of 1, 5, 10, 100, with the optimum ratio taken forward to subsequent rounds. The target

<sup>6</sup>Dynabench is an open-source platform which supports dynamic dataset generation and model benchmarking for a variety of NLP tasks (Kiela et al., 2021).

<sup>7</sup>There was no overlap in annotators from the previous HATEMOJICHECK task; demographics are in Appendix D.

model is then re-trained on the training data from all prior rounds as well as the current round. For R5 data collection, the target model (R5-T) is the DeBERTa model released by Ma et al. (2021), trained on R0–R4 from Vidgen et al. (2021b). This model performs well on text-only hate, but has seen limited emoji in training. For subsequent target models, we evaluate two architectures: DeBERTA (He et al., 2021) and BERTweet (Nguyen et al., 2020).<sup>8</sup>

We select the best models and upsampling ratios at the end of each round. The criteria to select the best model must satisfy two requirements: (1) the target model must still perform well on prior rounds of text-based examples because emoji-based hate represents just one construction of textual hate, and (2) the target model must successfully handle the most recent round data because iterative data collection produces more challenging examples in successive rounds. To balance these requirements, we use a weighted accuracy metric with 50% weight on the test sets from all prior rounds and 50% weight on the current round test set, which enforces a recency-based discount factor (Kiela et al., 2021). This ensures we can assess performance against the latest (emoji-specific and most-adversarial) round without overfitting and reducing performance on the previous test sets.

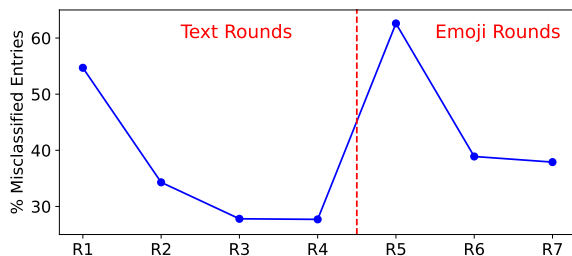


Figure 1: MER for R1–4 (text) and R5–7 (emoji).

**Model Error Rate** For each emoji-based round of data generation (R5–7) as well as prior text-only rounds from Vidgen et al. (2021b), we calculate model error rate (MER) as the proportion of annotators’ original entries that fool the model (Fig. 1). Between R4 and R5, MER increases by 35 percentage points (pp) to over 60%, higher than the R1 MER. From R5 to R6, once the model has been trained on emoji-based hate, there is a steep reduction in MER of 24pp. From R6 to R7, there is a smaller reduction in MER of 1pp. Overall, the

<sup>8</sup>For further details of model training, selection and emoji encodings see Appendix F.

model is easily tricked by emoji content at first but then becomes much harder to trick after our first round of re-training.

**Performance on Adversarial Test Sets** To evaluate model performance across rounds, we calculate F1-score and accuracy for each test set (Tab. 2). The baseline R5-T model is the highest performing model on the text-only test set (R1–4), with an F1 of 0.85, but only scores 0.49 on the newly generated emoji-containing test sets from HATEMOJIBUILD (R5–7). R6-T, R7-T and R8-T perform better on R5–7, with F1 between 0.76 and 0.77. They perform similarly well on the R1–4 test set, with F1 of 0.84, suggesting no trade-off for performance on text-only hate speech. The best performing model across all R1–7 test sets is R8-T, with an F1 of 0.83. The greatest performance gain in the adversarially-trained models is from R5-T to R6-T, with an increase in F1 of 0.28 on the emoji test set. This is achieved with only 2,000 emoji-containing examples. By comparison, R7-T and R8-T yield very small improvements.

Table 2: Performance of target models on emoji test sets from HATEMOJIBUILD, text sets from Vidgen et al. (2021b) and the combination of these test sets.

	Emoji Test Set (R5-R7) <i>n</i> = 593		Text Test Set (R1-R4) <i>n</i> = 4119		All Test Sets (R1-R7) <i>n</i> = 4712	
	Acc	F1	Acc	F1	Acc	F1
<b>R5-T</b>	0.585	0.490	<b>0.828</b>	<b>0.847</b>	0.786	0.801
<b>R6-T</b>	0.757	<b>0.769</b>	0.823	0.837	0.813	0.825
<b>R7-T</b>	<b>0.759</b>	0.762	0.824	0.842	0.813	0.829
<b>R8-T</b>	0.744	0.755	0.827	0.844	<b>0.814</b>	<b>0.829</b>

## 4 Evaluating Models with HATEMOJICHECK

We present results for two existing models as baselines, with additional baselines shown in Appendix G. The first baseline is Google Jigsaw’s Perspective API, a widely-used commercial tool for content moderation. We use Perspective’s “Identity Attack” attribute, which is defined as “negative or hateful comments targeting someone because of their identity” and thus closely matches our definition of hate. The returned score is converted to a binary label with a 50% cutoff. We refer to this model as P-IA. The second baseline is the R5-T model from Ma et al. (2021), introduced above. To compare model performance on HATEMOJICHECK, we use accuracy because three sets of test cases (originals, polarity perturbations and identity perturba-

Table 3: Aggregate accuracy for models evaluated on HATEMOJICHECK.

	<i>n</i>	Pre-Emoji		Post-Emoji		
		P-IA	R5-T	R6-T	R7-T	R8-T
<b>Overall</b>	3930	0.689	0.779	<b>0.879</b>	0.867	0.871
<b>Label</b>						
Hate	2654	0.706	0.770	<b>0.905</b>	0.876	0.896
Not hate	1276	0.653	0.799	0.825	<b>0.849</b>	0.820
<b>Set</b>						
Original	2126	0.664	0.717	<b>0.887</b>	0.850	0.877
Identity p.	314	0.908	0.917	<b>0.917</b>	0.917	0.876
Polarity p.	902	0.584	0.778	0.818	<b>0.853</b>	0.830
No emoji p.	588	0.823	0.934	<b>0.925</b>	0.925	0.910
<i>emoji diff</i>		-0.159	-0.217	-0.038	-0.075	<b>-0.033</b>

tions) have one class label, making F1-score incompatible. To measure emoji-specific weaknesses, we also calculate *emoji difference*, the difference between averaged model accuracy on the original emoji test cases compared with averaged accuracy on the no emoji perturbations.

On HATEMOJICHECK as a whole, our newly trained models R6-T, R7-T and R8-T perform best, with overall accuracy from 0.87 to 0.88 (Tab. 3). They substantially outperform P-IA, with an accuracy of 0.69, and R5-T, with an accuracy of 0.78. Our newly trained models have the smallest emoji difference, between 0.03 and 0.08. In contrast, P-IA and R5-T have emoji differences of 0.16 and 0.22, respectively. Comparing the three models trained on HATEMOJIBUILD, the first round of adversarial data generation yields the largest relative improvement, and in many ways R6-T is at least as good a model, if not better, than R8-T.

#### 4.1 Model Performance by Functionality

Our models trained on HATEMOJIBUILD perform better than the two baseline models on nearly every functionality (Tab. 4). They also perform far more consistently across all perturbation types.

For F1 Verb Swap, P-IA and R5-T perform well on original statements, but then perform poorly on the polarity perturbations (0.20 and 0.42 accuracy, respectively). Our models have much stronger performance on polarity perturbations (between 0.73 and 0.77), and comparably high performance on the other sets of test cases. For F2 Identity Swap, P-IA and R5-T perform very poorly on original statements (0.14 and 0.33 accuracy, respectively) but then perform well on the polarity perturbations and no emoji perturbations. This vulnerability is carried forward to performance on F4 Double Swap. R5-T only achieves 0.03 and P-IA makes zero correct

predictions. In contrast, our models achieve accuracy of 0.83 on F2 and accuracies between 0.70 and 0.79 on F4. For F3 Descriptor Swap, our models improve over P-IA and R5-T on the original statements. The relative improvement is particularly large for the polarity perturbations (0.25 for P-IA compared with 0.93 for R8-T). For F5 Append, P-IA and R5-T perform moderately with accuracies of 0.73 and 0.69. Our models perform far better, with accuracies from 0.87 to 0.99. However, they do not show a substantial increase in performance for the no emoji perturbations in F5.3. For F6 Positive Confounder, our models perform worse than P-IA and R5-T on the original statements, but then have far more consistent performance across the perturbations. For instance, R8-T achieves accuracies between 0.89 to 0.93 across all sets of test cases in this functionality, compared with a range of 0.44 to 0.93 for P-IA. P-IA and R5-T perform well on original statements and poorly on perturbations precisely because they ignore the effect of the emoji confounder. F7 Emoji Leetspeak is a successful adversarial strategy. R5-T achieves 0.85 accuracy on the original statements but just 0.67 on the identity perturbations, while P-IA does even worse, with only 0.59 on the original statements and 0.57 on the polarity perturbations. Our models perform better on the original statements, but still struggle with identity perturbations, showing how challenging this functionality is. However, non-hateful leetspeak constructions are likely less prevalent on social media than slur-based leetspeak.

Overall, all models trained on HATEMOJIBUILD perform well on emoji-based hate, but there is no clear ‘best’ model among them. After the large improvement from R5-T to R6-T, subsequent models produce minimal performance gains across the functionalities. This suggests the gains, as measured on HATEMOJICHECK’s intentionally short and simple statements, are quickly saturated.

#### 4.2 Model Performance by Target Group

HATEMOJICHECK contains target group labels for hateful and non-hateful entries, allowing models to be compared by their subgroup fairness. However, defining *fairness* is a non-trivial task, and there is substantial debate in the machine learning community as to which metrics are most appropriate. We consider six metrics calculated across subgroups: accuracy, precision, recall, false positive rate, false

Table 4: Accuracy across HATEMOJICHECK’s seven functionalities and their associated perturbation sets.

Functionality	Label	Example Test Case Template	n	Pre-Emoji		Post-Emoji		
				P-IA	R5-T	R6-T	R7-T	R8-T
<b>F1 Verb Swap</b>	<b>hate</b>	I want to / all [IDENTITY]	300	0.94	0.85	0.84	0.79	0.84
<b>F1.1</b> Identity perturbation	<b>not hate</b>	I want to / all bigots	50	0.94	1.00	1.00	0.98	0.86
<b>F1.2</b> Polarity perturbation	<b>not hate</b>	I want to 🙏 all [IDENTITY]	60	0.20	0.42	0.75	0.77	0.73
<b>F1.3</b> No emoji perturbation	<b>hate</b>	I want to kill all [IDENTITY]	60	1.00	0.95	0.97	0.97	0.98
<i>emoji diff:</i>				-0.06	-0.10	-0.12	-0.18	-0.15
<b>F2 Identity Swap</b>	<b>hate</b>	I hate [IDENTITY EMOJI]	120	0.14	0.33	0.83	0.83	0.83
<b>F2.1</b> Identity perturbation	<b>not hate</b>	I hate 🍕	20	0.90	0.70	1.00	0.85	1.00
<b>F2.2</b> Polarity perturbation	<b>not hate</b>	I love [IDENTITY EMOJI]	120	1.00	0.98	0.78	0.88	0.93
<b>F2.3</b> No emoji perturbation	<b>hate</b>	I hate [IDENTITY]	120	0.98	0.98	0.97	1.00	0.95
<i>emoji diff:</i>				-0.84	-0.66	-0.13	-0.17	-0.12
<b>F3 Descriptor Swap</b>	<b>hate</b>	[IDENTITY] are nothing but 🐭	260	0.92	0.83	0.99	0.99	1.00
<b>F3.1</b> Identity perturbation	<b>not hate</b>	Bullies are nothing but 🐭	40	1.00	1.00	1.00	1.00	1.00
<b>F3.2</b> Polarity perturbation	<b>not hate</b>	[IDENTITY] are nothing but ❤️	60	0.25	0.48	0.78	0.82	0.93
<b>F3.3</b> No emoji perturbation	<b>hate</b>	[IDENTITY] are nothing but vermin	60	0.98	1.00	1.00	1.00	1.00
<i>emoji diff:</i>				-0.06	-0.17	-0.01	-0.01	0.00
<b>F4 Double Swap</b>	<b>hate</b>	I will 🍷 [IDENTITY EMOJI]	288	0.00	0.03	0.79	0.70	0.77
<b>F4.1</b> Identity perturbation	<b>not hate</b>	I will 🍷🦋	46	1.00	1.00	0.91	0.98	0.91
<b>F4.2</b> Polarity perturbation	<b>not hate</b>	I will ❤️ [IDENTITY EMOJI]	60	1.00	0.98	0.92	0.85	0.92
<b>F4.3</b> No emoji perturbation	<b>hate</b>	I will hurt [IDENTITY]	60	0.97	1.00	1.00	1.00	1.00
<i>emoji diff:</i>				-0.96	-0.97	-0.21	-0.30	-0.23
<b>F5 Append</b>	<b>hate</b>	My new manager is [IDENTITY] 🙄	288	0.73	0.69	0.99	0.89	0.87
<b>F5.1</b> Identity perturbation	<b>not hate</b>	My new manager is a cat-lover 🐱	48	1.00	1.00	1.00	1.00	1.00
<b>F5.2</b> Polarity perturbation	<b>not hate</b>	My new manager is [IDENTITY] 👍	60	0.47	0.55	0.85	0.78	0.75
<b>F5.3</b> No emoji perturbation	<b>not hate</b>	My new manager is [IDENTITY]	60	0.35	0.48	0.45	0.43	0.37
<i>emoji diff:</i>				†	†	†	†	†
<b>F6 Positive Confounder</b>	<b>hate</b>	I find all [IDENTITY] disgusting 🤢	440	0.93	1.00	0.85	0.90	0.89
<b>F6.1</b> Identity perturbation	<b>not hate</b>	I find all reptiles disgusting 🦎	65	0.75	0.92	0.92	0.92	0.92
<b>F6.2</b> Polarity perturbation	<b>not hate</b>	I find all [IDENTITY] inspiring 🙌	112	0.44	0.88	0.96	0.91	0.93
<b>F6.3</b> No emoji perturbation	<b>hate</b>	I find all [IDENTITY] disgusting	88	0.93	0.99	0.95	1.00	0.93
<i>emoji diff:</i>				0.00	0.01	-0.10	-0.10	-0.04
<b>F7 Emoji Leetspeak</b>	<b>hate</b>	I hate [SLUR_EMOJI]	430	0.59	0.85	0.91	0.83	0.91
<b>F7.1</b> Identity perturbation	<b>not hate</b>	I hate d0gs	45	0.82	0.67	0.62	0.64	0.49
<b>F7.2</b> Polarity perturbation	<b>not hate</b>	I hate people who say [SLUR_EMOJI]	430	0.57	0.79	0.79	0.86	0.77
<b>F7.3</b> No emoji perturbation	<b>hate</b>	I hate [SLUR]	140	0.61	0.99	0.99	0.94	0.99
<i>emoji diff:</i>				-0.03	-0.14	-0.09	-0.11	-0.07

Notes: † The *emoji difference* is not defined because the no emoji perturbation is non-hateful, i.e., the opposite label to the original test case, so accuracy over these sets cannot be fairly compared.

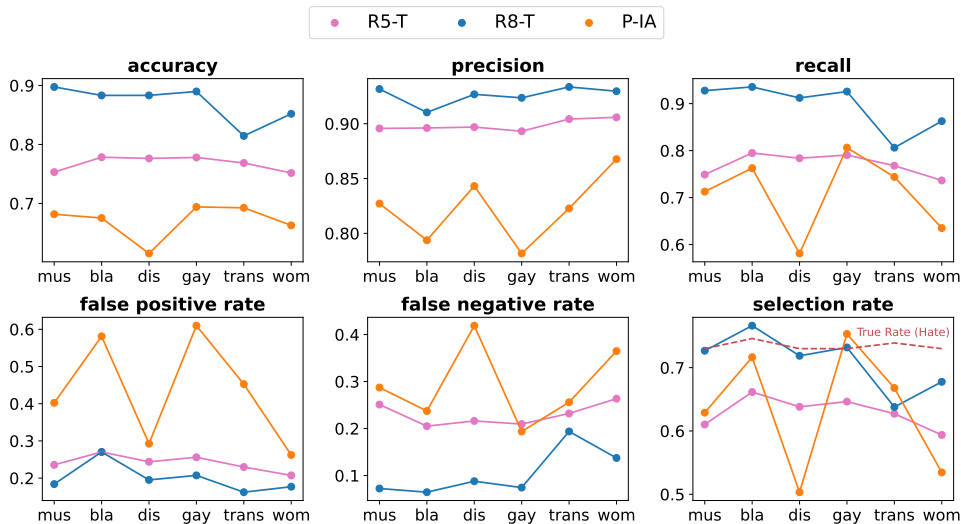


Figure 2: Subgroup fairness metrics by model. Abbreviations respectively refer to Muslims, Black people, Disabled people, Gay people, Trans people and Women.

negative rate and selection rate.<sup>9</sup> Fig. 2 shows subgroup fairness metrics compared across three models: P-IA, R5-T and R8-T. It is concerning that P-IA has such differential performance, especially regarding the unbalanced false positive and negative rates for women and disabled people, as both of these error types are societally harmful. R8-T has more balanced accuracy, precision, recall and selection rate across subgroups, driven by stable performance in false negatives and positives. In Appendix H we provide two further between-group fairness metrics: demographic parity ratio (Agarwal et al., 2019), and equalized odds ratio (Hardt et al., 2016; Agarwal et al., 2018). There is also substantial improvement in these metrics from adversarial training schemes relative to commercial solutions or statically-trained models.

## 5 Discussion

HATEMOJICHECK reveals critical model weaknesses in detecting emoji-based hate. Existing commercial and academic models perform poorly at identifying hate where the identity term has been replaced with an emoji representation (F2 and F4), even though they perform well at identifying the equivalent textual statements. These models have better performance on Verb Swap (F1) but then struggle with the polarity perturbations (F1.2). This suggests that the models are overfitting to the identity term and ignore the sentiment from the emoji, leading to false positive predictions. Our newly trained models substantially improve performance on original hateful statements from F2, F4 and F5, indicating they have a better semantic grasp of emoji substitutions and appends. They also make large performance gains on the polarity perturbations in F1.2, F3.2, F5.2 and F6.2, suggesting they better incorporate information on how different emoji condition the likelihood of hatefulness. Despite improving on existing models, our models still perform relatively more poorly on F4, as well as the F1.2, F5.3 and F7.1 perturbations. These weaknesses could potentially be addressed in future work through additional data generation.

Training on just one round of adversarial data yields the biggest improvement on HATEMOJICHECK. Thereafter, performance plateaus. This aligns with the sharp increase and then fall in MER

across emoji-based rounds of data generation (in Fig. 1). On the adversarial test sets, R8-T only marginally outperforms R6-T and R7-T on all R1–7 test sets. It slightly underperforms on the R5–7 test sets from HATEMOJIBUILD. This suggests that while training on a relatively small number of entries can substantially improve performance, the gains quickly saturate. Future work could investigate performance differences in more detail using the labels for type and target of hate that we provide for all three rounds of data.

Due to practical constraints, HATEMOJICHECK has several limitations, which could be addressed in future work. First, it includes relatively simplistic statements, and so offers negative predictive power: high performance only indicates the absence of specific weaknesses (Gardner et al., 2020; Röttger et al., 2021). Second, it is inspired by real-world hate but synthetically generated. Future work could evaluate against more complex and diverse forms of emoji-based hate. Third, it is limited in scope. It only considers short English-language statements with one binary label. Only six identities are included, none of which are intersectional, and the set of emoji covers a fraction of the full Unicode Standard. In-scope identities and emoji usage are culturally-dependent (Barbieri et al., 2016; Ljubešić and Fišer, 2016) so future work could assess the generalizability of HATEMOJICHECK to other cultures and languages. These limitations do not diminish HATEMOJICHECK’s utility as a tool for effectively identifying model weaknesses to emoji-based hate. By publicly releasing the test suite, practitioners and academics can scrutinize their models prior to deployment or publication.

Adversarial data generation is a powerful technique for creating diverse, complex and informative datasets. However, it also introduces challenges. First, the entries are ‘synthetic’ rather than sampled from ‘real-world’ examples. Substantial training and time are required to ensure that annotators understand real online hate and can imitate it. Second, annotators can exhaust their creativity and start producing unrealistic, simplistic or non-adversarial examples. Third, because of the need for training, supervision and support, only a small pool of annotators is feasible, which can introduce additional idiosyncratic biases. Given these issues, quality control is a key consideration throughout the data generation process. Encouragingly, we find carefully-curated and adversarially-generated

<sup>9</sup>Selection rate is  $\mathbb{P}[h(x)|A = a]$ , i.e., the proportion of the population in subgroup  $a$  who have 1 (hate) as the given class label. The ‘true’ ratio of class labels is shown in Fig. 2



training datasets can significantly improve performance on emoji-based hate as a particular type of challenging content, and that this approach is effective with relatively few training examples. Thus, substantial model improvements can be realized with minimal financial and computational cost.

## 6 Related Work

Emoji-based hate has received limited attention in prior work on hate and abusive language detection. Studies that attend to emoji often do so as a potential input feature to aid classifier performance. For example, [Samghabadi et al. \(2019\)](#) improve offensive language classification with an ‘emotion-aware’ mechanism built on emoji embeddings. [Ibrohim et al. \(2019\)](#) find that adding emoji features to Continuous Bag of Words and word unigram models marginally improves performance for abusive language detection on Indonesian Twitter. [Bick \(2020\)](#) identifies examples of subtle and non-direct hate speech in German–Danish Twitter conveyed through ‘winking’ or ‘skeptical’ emoji that flag irony or non-literal meaning in their accompanying text. [Corazza et al. \(2020\)](#) train an emoji-based Masked Language Model (MLM) for zero-shot abuse detection. They show this method improves performance on classifying abuse in German, Italian and Spanish tweets compared to an MLM which does not attend to emoji. [Wiegand and Ruppenhofer \(2021\)](#) use abusive emoji as a proxy for learning a lexicon of abusive words. Their findings indicate that emoji can disambiguate abusive and profane usages of words such as f\*ck and b\*tch. By contrast, our work focuses on emoji-based hate as a challenge for hate detection models. With HATEMOJICHECK, we enable a systematic evaluation of how well models handle different types of emoji-based hate. Rather than adjusting the model architecture, we account for emoji-based hate in our iterative data generation process and show that models trained on such data perform better on emoji-based hate, while retaining strong performance on text-only hate.

As a suite of functional tests for evaluation, HATEMOJICHECK directly builds on previous work by [Ribeiro et al. \(2020\)](#) and [Röttger et al. \(2021\)](#). [Ribeiro et al. \(2020\)](#) introduced functional tests as a framework for NLP model evaluation with CHECKLIST, showing that their approach can identify granular model strengths and weaknesses that are obscured by high-level metrics like accuracy

and F1-score. [Röttger et al. \(2021\)](#) adapted this framework to hate detection with HATECHECK, which covers 29 model functionalities motivated by interviews with civil society stakeholders and a review of previous hate speech literature. Like HATECHECK, we pair hateful test cases with contrasting perturbations that are particularly challenging to models relying on overly simplistic decision rules and thus reveal granular decision boundaries. HATECHECK did not consider emoji-based hate, which is the main focus of our work.

Our approach to training better models for emoji-based hate builds directly on work by [Vidgen et al. \(2021b\)](#), who apply iterative human-and-model-in-the-loop training to hate detection models. Like us, they used the Dynabench interface ([Kiela et al., 2021](#)) to implement their training system, which has also been used to improve model performance for other tasks such as reading comprehension ([Bartolo et al., 2020](#)) and sentiment analysis ([Potts et al., 2021](#)). Earlier work by [Dinan et al. \(2019\)](#) introduced a similar ‘build it, break it, fix it’ system of repeated interactions between a hate classifier and crowdworkers to develop safe-by-design chatbots. Unlike previous work, we focus data generation on a particular type of hateful content, emoji-based hate, and show that the training scheme can address specific model weaknesses on such content without sacrificing performance on text-only hate.

## 7 Conclusion

Online hate is a pervasive, harmful phenomenon, and hate detection models are a crucial tool for tackling it at scale. We showed that emoji pose a particular challenge for such models, and presented HATEMOJICHECK, a first-of-its-kind evaluation suite for emoji-based hate. It covers seven functionalities with 3,930 test cases. Using this test suite, we exposed clear weaknesses in the performance of commercial and academic models on emoji-based hate. To address these weaknesses, we created the HATEMOJIBUILD dataset using an innovative human-and-model-in-the-loop approach. We showed that models trained on this adversarial data are substantially better at detecting emoji-based hate, while retaining strong performance on text-only hate. Our approach of first identifying granular model weaknesses, and then generating an adversarial dataset to address them presents a promising direction for building models to detect other diverse and emerging forms of online harm.

## Acknowledgments

We are thankful for support that the Oxford authors received to facilitate annotation and computational resources from the Volkswagen Foundation, Meedan, Keble College, the Oxford Internet Institute, Rewire and The Alan Turing Institute. We owe a debt of gratitude to all our annotators, to Dynabench and to our anonymous reviewers. We are also grateful to Zeerak Talat and Douwe Kiela for their helpful advice on this research, as well as support from Devin Gaffney and Darius Kazemi. Hannah Rose Kirk was supported by the Economic and Social Research Council grant ES/P000649/1. Paul Röttger was supported by the German Academic Scholarship Foundation.

## References

- Alekh Agarwal, Alina Beygelzimer, Miroslav Dudík, John Langford, and Hanna Wallach. 2018. [A Reductions Approach to Fair Classification](#). *arXiv:1803.02453 [cs]*.
- Alekh Agarwal, Miroslav Dudík, and Zhiwei Steven Wu. 2019. Fair Regression: Quantitative Definitions and Reduction-Based Algorithms. In *Proceedings of the 36th International Conference on Machine Learning*, pages 120–129. PMLR.
- Francesco Barbieri, German Kruszewski, Francesco Ronzano, and Horacio Saggion. 2016. [How Cosmopolitan Are Emojis? Exploring Emojis Usage and Meaning over Different Languages with Distributional Semantics](#). In *Proceedings of the 24th ACM International Conference on Multimedia*, MM '16, pages 531–535, New York, NY, USA. Association for Computing Machinery.
- Max Bartolo, Alastair Roberts, Johannes Welbl, Sebastian Riedel, and Pontus Stenetorp. 2020. [Beat the AI: Investigating adversarial human annotation for reading comprehension](#). *Transactions of the Association for Computational Linguistics*, 8:662–678.
- Anna Bax. 2018. [“The C-Word” Meets “the N-Word”: The Slur-Once-Removed and the Discursive Construction of “Reverse Racism”](#). *Journal of Linguistic Anthropology*, 28(2):114–136.
- Boris Beizer. 1995. *Black-Box Testing: Techniques for Functional Testing of Software and Systems*. John Wiley & Sons, Inc., New York, NY, USA.
- Emily M. Bender and Batya Friedman. 2018. [Data statements for natural language processing: Toward mitigating system bias and enabling better science](#). *Transactions of the Association for Computational Linguistics*, 6:587–604.
- Eckhard Bick. 2020. Annotating emoticons and emojis in a german-danish social media corpus for hate speech research. *RASK*, 52.
- Brandwatch. 2018. [The Emoji Report](#).
- Spencer Cappallo, Stacey Svetlichnaya, Pierre Garrigues, Thomas Mensink, and Cees G. M. Snoek. 2019. [New Modality: Emoji Challenges in Prediction, Anticipation, and Retrieval](#). *IEEE Transactions on Multimedia*, 21(2):402–415.
- Michele Corazza, Stefano Menini, Elena Cabrio, Sara Tonelli, and Serena Villata. 2020. [Hybrid Emoji-Based Masked Language Models for Zero-Shot Abusive Language Detection](#). In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 943–949, Online. Association for Computational Linguistics.
- Juliet M. Corbin and Anselm Strauss. 1990. [Grounded theory research: Procedures, canons, and evaluative criteria](#). *Qualitative Sociology*, 13(1):3–21.
- Thomas Davidson, Dana Warmusley, Michael Macy, and Ingmar Weber. 2017. Automated Hate Speech Detection and the Problem of Offensive Language. *Proceedings of the International AAAI Conference on Web and Social Media*, 11(1):512–515.
- Leon Derczynski, Hannah Rose Kirk, Abeba Birhane, and Bertie Vidgen. 2022. [Handling and Presenting Harmful Text](#). *arXiv:2204.14256 [cs]*.
- Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. [BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding](#). In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.
- Emily Dinan, Samuel Humeau, Bharath Chintagunta, and Jason Weston. 2019. [Build it Break it Fix it for Dialogue Safety: Robustness from Adversarial Human Attack](#). 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 4537–4546, Hong Kong, China. Association for Computational Linguistics.
- Lucas Dixon, John Li, Jeffrey Sorensen, Nithum Thain, and Lucy Vasserman. 2018. [Measuring and Mitigating Unintended Bias in Text Classification](#). In *Proceedings of the 2018 AAAI/ACM Conference on AI, Ethics, and Society*, AIES '18, pages 67–73, New York, NY, USA. Association for Computing Machinery.
- Bjarke Felbo, Alan Mislove, Anders Søgaard, Iyad Rahwan, and Sune Lehmann. 2017. [Using millions of emoji occurrences to learn any-domain representations for detecting sentiment, emotion and sarcasm](#).

- In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1615–1625, Copenhagen, Denmark. Association for Computational Linguistics.
- Antigoni-Maria Founta, Constantinos Djouvas, Despoina Chatzakou, Ilias Leontiadis, Jeremy Blackburn, Gianluca Stringhini, Athena Vakali, Michael Sirivianos, and Nicolas Kourtellis. 2018. **Large Scale Crowdsourcing and Characterization of Twitter Abusive Behavior**. In *Twelfth International AAAI Conference on Web and Social Media*.
- Matt Gardner, Yoav Artzi, Victoria Basmov, Jonathan Berant, Ben Bogin, Sihao Chen, Pradeep Dasigi, Dheeru Dua, Yanai Elazar, Ananth Gottumukkala, Nitish Gupta, Hannaneh Hajishirzi, Gabriel Ilharco, Daniel Khoshabi, Kevin Lin, Jiangming Liu, Nelson F. Liu, Phoebe Mulcaire, Qiang Ning, Sameer Singh, Noah A. Smith, Sanjay Subramanian, Reut Tsarfaty, Eric Wallace, Ally Zhang, and Ben Zhou. 2020. **Evaluating Models’ Local Decision Boundaries via Contrast Sets**. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 1307–1323, Online. Association for Computational Linguistics.
- Katharine Gelber and Luke McNamara. 2016. **Evidencing the harms of hate speech**. *Social Identities*, 22(3):324–341.
- Tarleton Gillespie. 2020. **Content moderation, AI, and the question of scale**. *Big Data and Society*, 7(2).
- Moritz Hardt, Eric Price, and Nati Srebro. 2016. **Equality of Opportunity in Supervised Learning**. In *Advances in Neural Information Processing Systems*, volume 29. Curran Associates, Inc.
- Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. 2021. **DeBERTa: Decoding-enhanced BERT with Disentangled Attention**. *arXiv:2006.03654 [cs]*.
- Muhammad Okky Ibrohim, Muhammad Akbar Setiadi, and Indra Budi. 2019. **Identification of hate speech and abusive language on Indonesian Twitter using the Word2vec, part of speech and emoji features**. In *Proceedings of the International Conference on Advanced Information Science and System, AISS ’19*, pages 1–5, New York, NY, USA. Association for Computing Machinery.
- Alastair Jamieson. 2020. **Racist comments after Euro 2020: Saka, Sancho and Rashford racially abused online after England defeat**.
- Douwe Kiela, Max Bartolo, Yixin Nie, Divyansh Kaushik, Atticus Geiger, Zhengxuan Wu, Bertie Vidgen, Grusha Prasad, Amanpreet Singh, Pratik Ringshia, Zhiyi Ma, Tristan Thrush, Sebastian Riedel, Zeerak Waseem, Pontus Stenetorp, Robin Jia, Mohit Bansal, Christopher Potts, and Adina Williams. 2021. **Dynabench: Rethinking Benchmarking in NLP**. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4110–4124, Online. Association for Computational Linguistics.
- J. Richard Landis and Gary G. Koch. 1977. **The Measurement of Observer Agreement for Categorical Data**. *Biometrics*, 33(1):159–174.
- Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. **RoBERTa: A Robustly Optimized BERT Pretraining Approach**. *arXiv:1907.11692 [cs]*.
- Nikola Ljubešić and Darja Fišer. 2016. **A global analysis of emoji usage**. In *Proceedings of the 10th Web as Corpus Workshop*, pages 82–89, Stroudsburg, PA, USA. Association for Computational Linguistics.
- Zhiyi Ma, Kawin Ethayarajh, Tristan Thrush, Somya Jain, Ledell Wu, Robin Jia, Christopher Potts, Adina Williams, and Douwe Kiela. 2021. **Dynaboard: An Evaluation-As-A-Service Platform for Holistic Next-Generation Benchmarking**. In *Advances in Neural Information Processing Systems*, volume 34, pages 10351–10367. Curran Associates, Inc.
- Dat Quoc Nguyen, Thanh Vu, and Anh Tuan Nguyen. 2020. **BERTweet: A pre-trained language model for English Tweets**. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 9–14, Online. Association for Computational Linguistics.
- Christopher Potts, Zhengxuan Wu, Atticus Geiger, and Douwe Kiela. 2021. **DynaSent: A Dynamic Benchmark for Sentiment Analysis**. 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 2388–2404, Online. Association for Computational Linguistics.
- Justas Randolph. 2005. **Free-marginal multirater kappa: An alternative to fleiss**. In *Joensuu Learning and Instruction Symposium*, page 20.
- Marco Tulio Ribeiro, Tongshuang Wu, Carlos Guestrin, and Sameer Singh. 2020. **Beyond Accuracy: Behavioral Testing of NLP Models with CheckList**. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4902–4912, Online. Association for Computational Linguistics.
- David Rodrigues, Marília Prada, Rui Gaspar, Margarida V. Garrido, and Diniz Lopes. 2018. **Lisbon Emoji and Emoticon Database (LEED): Norms for emoji and emoticons in seven evaluative dimensions**. *Behavior Research Methods*, 50(1):392–405.
- Paul Röttger, Bertie Vidgen, Dirk Hovy, and Janet B. Pierrehumbert. 2022. **Two Contrasting Data Annotation Paradigms for Subjective NLP Tasks**. *arXiv:2112.07475 [cs]*.

- Paul Röttger, Bertie Vidgen, Dong Nguyen, Zeerak Waseem, Helen Margetts, and Janet Pierrehumbert. 2021. [HateCheck: Functional Tests for Hate Speech Detection Models](#). In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 41–58, Online. Association for Computational Linguistics.
- Niloofer Safi Samghabadi, Afsheen Hatami, Mahsa Shafaei, Sudipta Kar, and Tamar Solorio. 2019. Attending the emotions to detect online abusive language.
- Leandro Silva, Mainack Mondal, Denzil Correa, Fabricio Benevenuto, and Ingmar Weber. 2016. Analyzing the targets of hate in online social media. *arXiv*.
- Zeerak Talat, Thomas Davidson, Dana Warmusley, and Ingmar Weber. 2017. [Understanding Abuse: A Typology of Abusive Language Detection Subtasks](#). In *Proceedings of the First Workshop on Abusive Language Online*, pages 78–84, Vancouver, BC, Canada. Association for Computational Linguistics.
- United Nations. 2019. UN strategy and plan of action on hate speech. Technical report.
- Bertie Vidgen, Alex Harris, Dong Nguyen, Rebekah Tromble, Scott Hale, and Helen Margetts. 2019. [Challenges and frontiers in abusive content detection](#). In *Proceedings of the Third Workshop on Abusive Language Online*, pages 80–93, Florence, Italy. Association for Computational Linguistics.
- Bertie Vidgen, Dong Nguyen, Helen Margetts, Patricia Rossini, and Rebekah Tromble. 2021a. [Introducing CAD: The Contextual Abuse Dataset](#). In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2289–2303, Online. Association for Computational Linguistics.
- Bertie Vidgen, Tristan Thrush, Zeerak Waseem, and Douwe Kiela. 2021b. [Learning from the Worst: Dynamically Generated Datasets to Improve Online Hate Detection](#). 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 1667–1682, Online. Association for Computational Linguistics.
- Michael Wiegand and Josef Ruppenhofer. 2021. [Exploiting Emojis for Abusive Language Detection](#). In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 369–380, Online. Association for Computational Linguistics.
- Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi 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 M. Rush. 2020. [HuggingFace’s Transformers: State-of-the-art Natural Language Processing](#). *arXiv:1910.03771 [cs]*.
- Marcos Zampieri, Shervin Malmasi, Preslav Nakov, Sara Rosenthal, Noura Farra, and Ritesh Kumar. 2019. [SemEval-2019 Task 6: Identifying and Categorizing Offensive Language in Social Media \(OffensEval\)](#). In *Proceedings of the 13th International Workshop on Semantic Evaluation*, pages 75–86, Minneapolis, Minnesota, USA. Association for Computational Linguistics.
- Xiang Zhou, Yixin Nie, Hao Tan, and Mohit Bansal. 2020. [The Curse of Performance Instability in Analysis Datasets: Consequences, Source, and Suggestions](#). In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 8215–8228, Online. Association for Computational Linguistics.
- Yukun Zhu, Ryan Kiros, Richard Zemel, Ruslan Salakhutdinov, Raquel Urtasun, Antonio Torralba, and Sanja Fidler. 2015. [Aligning Books and Movies: Towards Story-like Visual Explanations by Watching Movies and Reading Books](#). *arXiv:1506.06724 [cs]*.

## A Ethical Considerations

**Misuse:** We release two datasets of challenging emoji examples on which commercial solutions and state-of-the-art transformer models have been proven to fail. Malicious actors could take inspiration for bypassing current detection systems on internet platforms, or in principal train a generative hate speech model. This concern is also raised by Vidgen et al. (2021b), but the conclusion is reached that such risk of misuse is small and outweighed by the considerable scientific and social benefits.

**Harm Statement:** Following the advice of Derczynski et al. (2022), we describe the risks to well-being during the production and publication of this research and the steps taken to mitigate them. There is a risk of harm to data subjects i.e., the targets of hate, from reinforcing hateful, dehumanizing or derogatory statements, and to readers viewing this content in the paper. To mitigate these harms, we include a content warning directly after the abstract, at least a page before any harmful content is displayed and colored in red for maximum visibility. We also include a section-specific content warning before §2.2, which includes a number of hateful examples. Hateful examples in §2.2 are consistently formatted and visually distanced with gray text. All other examples (including Tab. 1, Tab. 4) are presented with placeholders for the [IDENTITY]. Hateful slurs are starred out with an asterisk where possible (e.g., Tab. 5) but we cannot star out emoji without hindering interpretation. There is also a risk to researchers and annotators from labeling and viewing harmful content. As authors, we oppose the use of hateful language. We follow protocols for protecting annotator well-being, including briefing sessions, regular check-ins and provision of mental health support.

## B Data Statement for HATEMOJICHECK

We provide a data statement (Bender and Friedman, 2018) to document the generation and provenance of HATEMOJICHECK.

### B.1 Curation Rationale

To construct HATEMOJICHECK, we hand-crafted 3,930 short-form English-language texts using a template-based method for group identities and slurs. Each test case exemplifies one functionality and is associated with a binary gold standard label (hate versus not hate). All 3,930 cases were labeled

by a trained team of three annotators, who could also flag examples that were unrealistic. Any test cases with multiple disagreements or flags were replaced with alternative templates and re-issued for annotation to improve the quality of examples in the final set of test cases. The purpose of HATEMOJICHECK is to evaluate the performance of black-box models against varied constructions of emoji-based hate.

### B.2 Language Variety

The test cases are in English. This choice was motivated by the researchers’ and annotators’ expertise, and to maximize HATEMOJICHECK’s applicability to previous hate speech detection studies, which are predominantly conducted on English-language data. We discuss the limitations of restricting HATEMOJICHECK to one language and suggest that future work should prioritize expanding the test suite to other languages.

### B.3 Speaker Demographics

All test cases were hand-crafted by the lead author, who is a native English-speaking researcher at a UK university with extensive subject matter expertise in online harms.

### B.4 Annotator Demographics

We recruited a team of three annotators who worked for two weeks in May 2021 and were paid £16/hour. All annotators were female and between 30–39 years old. One had an undergraduate degree, one a taught graduate degree and one a post-graduate research degree. There were three nationalities: Argentinian, British and Iraqi, two ethnicities: White and Arab, and three religious affiliations: Catholic, Muslim and None. One annotator was a native English speaker and the others were non-native but fluent. All annotators used emoji and social media more than once per day. All annotators had seen others targeted by abuse online, and one had been targeted personally.

### B.5 Speech Situation

The modality of all test cases is written text embedded with emoji. The empirical investigation using Twitter’s streaming API was conducted between 1st–14th April 2021. The first set of test cases was created between 26th April–7th May 2021. The first round of annotation ran between 7th–14th May 2021. The second round of cases was created and re-issued between 14th–21st May 2021.

## B.6 Text Characteristics

The genre of texts is hateful and non-hateful statements using emoji constructions. Renderings of emoji vary by operating system and browser providers. The renderings in this paper are from WhatsApp. The composition of the dataset by labels (hate versus not hate) and by set (originals versus perturbations) is described in Tab. 3 of the main paper. 68% of the test cases are hateful. HATEMOJICHECK has 644 cases for Muslims, 608 cases each for gay people, disabled people and women, 582 cases for Black people and 566 cases for trans people. identity perturbations switch a protected identity for a non-protected identity so 314 cases have no protected identity tag.

## C Constructing HATEMOJICHECK

### C.1 List of Hateful Slurs in HATEMOJICHECK

The slurs used in HATEMOJICHECK are selected in the same way as for HATECHECK (Röttger et al., 2021). For each of the six identities in HATEMOJICHECK, the most common slurs are taken from [hatebase.org](https://hatebase.org), an online crowd-sourced repository of hateful terms and slurs. The top three most common slurs logged by users of hatebase are used, unless the 3rd ranked slur is significantly less common than the 2nd ranked. Each identity thus has two or three associated slurs (Tab. 5).

### C.2 Defining Perturbations

Each template has three perturbations but the mapping of original test cases to perturbations is not one-to-one. Instead, there is a set of original templates contrasted by three sets of perturbation templates. The perturbation templates have fewer degrees of freedom than the original templates so test cases generated from them are fewer in number. Within each perturbed template we use the same emoji for substitutions to ensure consistency, but across templates, we use a range of emoji to ensure linguistic variety.

The perturbed templates are constructed as follows. **(1) Identity perturbations:** the protected identity (which could be an emoji or a word depending on the functionality) in each original template is substituted with *one* non-protected entity (which could be an emoji or a word). **(2) Polarity perturbations:** the negative term (which could be a word or an emoji) in each original template is substituted for *one* positive term (which could be a word or

Table 5: Hateful slurs in HATEMOJICHECK

Protected Group	Slurs
Women	b*tch, sl*t, wh*re
Trans people	tr*nny, sh*male
Gay people	f*ggot, f*g, q*eer
Black people	n*gger, c*on
Disabled people	r*tard, cr*ppl, m*ng
Muslims	m*zzi, J*hadi, camelf*cker

an emoji). **(3) No emoji perturbations:** all emoji elements of the original template are replaced by equivalent text where *one* word is used to cover all versions e.g. 🏳️ and 🏳️ are both substituted for the verb ‘harm’. For F5 and F6, the appended emoji is removed not replaced.

## D Data Statement for HATEMOJIBUILD

We provide a data statement (Bender and Friedman, 2018) to document the generation and provenance of HATEMOJIBUILD.

### D.1 Curation Rationale

We use an online interface designed for dynamic dataset generation and model benchmarking (Dynabench) to collect synthetic adversarial examples in three successive rounds, running between 24th May–11th June. Each round contains ~2,000 entries, where each original entry inputted to the interface is paired with an offline perturbation. Data was synthetically-generated by a team of trained annotators, i.e., not sampled from social media.

### D.2 Language Variety

All entries are in English. Language choice was dictated by the expertise of researchers and annotators. Furthermore, English is used for a wide number of benchmark hate speech datasets (Davidson et al., 2017; Founta et al., 2018) and was also used in the adversarial dataset for textual hate speech (Vidgen et al., 2021b). The method could be adapted for other languages in future work.

### D.3 Speaker Demographics

All entries are synthetically-created by annotators so the speaker demographics match the annotator demographics.

### D.4 Annotator Demographics

Ten annotators were recruited to work for three weeks, and paid £16/hour. An expert annotator was recruited for quality control purposes and paid £20/hour. In total, there were 11 annotators. All

annotators received a training session prior to data collection and had previous experience working on hate speech projects. A daily ‘stand-up’ meeting was held every morning to communicate feedback and update guidelines as rounds progressed. Annotators were able to contact the research team at any point using a messaging platform. Of 11 annotators, 8 were between 18–29 years old and 3 between 30–39 years old. The completed education level was high school for 3 annotators, undergraduate degree for 1 annotator, taught graduate degree for 4 annotators and post-graduate research degree for 3 annotators. 6 annotators were female, and 5 were male. Annotators came from a variety of nationalities, with 7 British, as well as Jordanian, Irish, Polish and Spanish. 7 annotators identified as ethnically White and the remaining annotators came from various ethnicities including Turkish, Middle Eastern, and Mixed White and South Asian. 4 annotators were Muslim, and others identified as Atheist or as having no religious affiliation. 9 annotators were native English speakers and 2 were non-native but fluent. The majority of annotators (9) used emoji and social media more than once per day. 10 annotators had seen others targeted by abuse online, and 7 had been personally targeted.

### D.5 Speech Situation

Entries were created from 24th May–11th June 2021. Their modality is short-form written texts embedded with emoji. Entries are synthetically-generated but annotators were trained on real-world examples of emoji-based hate from Twitter.

### D.6 Text Characteristics

The genre of texts is hateful and non-hateful statements using emoji constructions. Annotators inputted emoji into the platform using a custom emoji picker.<sup>10</sup> The composition of the final dataset is described in Tab. 6. 50% of the 5,912 test cases are hateful. 50% of the entries in the dataset are original content and 50% are perturbations.

## E Constructing HATEMOJIBUILD

**Types of Hate** We adopt the same categorization used by Vidgen et al. (2021b, p.3).<sup>11</sup> There are four types of hate. Derogation: Language which explicitly derogates, demonizes, demeans or insults

<sup>10</sup><https://hateemoji.stackblitz.io/>

<sup>11</sup>One of their categories “support for hateful entities” is excluded because it introduced confusion and ambiguity.

Table 6: Summary statistics across three rounds of data from HATEMOJIBUILD.

		R5	R6	R7
<i>n</i>		1994	1966	1952
Split, <i>n</i> (%)	Train	1595 (80.0)	1572 (80.0)	1561 (80.0)
	Dev	199 (10.0)	197 (10.0)	195 (10.0)
	Test	200 (10.0)	197 (10.0)	196 (10.0)
Label, <i>n</i> (%)	Hate	1006 (50.5)	983 (50.0)	976 (50.0)
	Not hate	988 (49.5)	983 (50.0)	976 (50.0)
Set, <i>n</i> (%)	Original	997 (50.0)	983 (50.0)	976 (50.0)
	Perturbation	997 (50.0)	983 (50.0)	976 (50.0)
Type, <i>n</i> (%)	None	988 (49.5)	983 (50.0)	976 (50.0)
	Derogation	718 (36.0)	649 (33.0)	594 (30.4)
	Animosity	74 (3.7)	219 (11.1)	275 (14.1)
	Threatening	101 (5.1)	50 (2.5)	52 (2.7)
	Dehumanizing	113 (5.7)	65 (3.3)	55 (2.8)
# Emoji, $\mu$ ( $\sigma$ )		1.7 (2.2)	1.7 (1.0)	1.6 (1.1)

a group. Animosity: Expressions of abuse through implicit statements or mockery, where a logical step must be taken between the sentence and its intended negativity. Threatening language: Statements of intent to take action against a group, with the potential to inflict serious or imminent harm on its members. Dehumanizing language: Comparing groups to insects, animals, germs or trash.

**Targets of Hate** Annotators were provided with a non-exhaustive list of high-priority identities to focus on which included categorizations by gender identity (e.g., women, trans), sexual orientation (e.g., gay), ethnicity (e.g., Hispanic people), religion (e.g., Sikh), nationality (e.g., Polish), disability and class, alongside intersections (e.g., Muslim women). Hate directed towards majority groups (e.g., men, white people and heterosexuals) is outside the remit of this work. The explicit decision not to focus on issues such as ‘reverse racism’ (Bax, 2018) is made due to the complex debate on its inclusion in hate speech definitions.

**Composition** Tab. 7 reports the inter-annotator agreement and number of final entries per round. Each round has approximately the same number of entries with slightly fewer in R7 due to more quality control issues. Labels are equally distributed in each round (see Tab. 6). For 75 pairs of originals and perturbations, the perturbation unsuccessfully flipped the label given by majority agreement between three annotators. All other pairs have opposite labels. Derogation is always the most-

Table 7: Quality control checks per round of data collection, showing Randolph-kappa score across three annotators per entry  $R(S)$ , entries reviewed by an expert annotator, and dropped entries.

		Input Entries	$R(S)$	Reviewed	Dropped	Final Entries
<b>R5</b>	Original	1020	0.903	273	6	1014
	Perturbation	1014	0.938	131	17	<b>997</b>
<b>R6</b>	Original	1008	0.902	189	8	1000
	Perturbation	1000	0.911	167	17	<b>983</b>
<b>R7</b>	Original	993	0.913	169	12	981
	Perturbation	981	0.928	168	5	<b>976</b>

commonly inputted form of hate. From R5 to R7, there is a rise in animosity entries paired with a decline in threatening and dehumanizing language entries. Annotators were given substantial freedom in the targets of hate resulting in 54 unique targets, and 126 unique intersections of these. The entries from R5–R7 contain 1,082 unique emoji out of 3,521 defined in the Unicode Standard as of September 2020. The mode of emoji per entries is 1 and mean is approximately 1.5 in each round. The frequency of targets and emoji follow a long-tailed distribution, similar to a Zipf curve. These distributions match those found online for targets (Silva et al., 2016) and for emoji (Felbo et al., 2017; Cappallo et al., 2019; Bick, 2020).

In the first round, annotators commonly employed a strategy of substituting identities for emoji so three identity emoji were frequently used (👤, 👥, 🌐). Using emoji character substitutions in slurs was also a successful strategy (A: 🅐), as was substituting the entire slur for an emoji homonym (f\*g: 🍌). In the final round, the model was wise to such strategies, and so annotators changed their adversarial techniques to irony, satire or mockery, as reflected in the top emoji (😏, 😊, 🤔, 🤪). For a qualitative understanding of annotators’ experiences, we conducted a post-study survey.<sup>12</sup>

## F Target Models

In each round we assessed two candidate model architectures. The first is an uncased DeBERTa base model (134M params) with a sequence classification head (He et al., 2021). DeBERTa has been shown to improve on BERT (Devlin et al., 2019) and RoBERTa (Liu et al., 2019) models by incorporating a disentangled attention mechanism and enhanced mask decoder. However, emoji are likely relatively sparse in DeBERTa’s pre-training

material which includes English Wikipedia, Book Corpus (Zhu et al., 2015) and a subset of Common-Crawl. DeBERTa uses the BPE-encoding method so the tokenizer can encode emoji as unique tokens. The second candidate is an uncased BERTweet model (135M params) with a sequence classification head where the RoBERTa training procedure was repeated on 850M English Tweets using a custom vocabulary (Nguyen et al., 2020). BERTweet tokenizes emoji by first translating them to text representations using the emoji package. These word representations are derived from the Unicode Standard, so for example 🤔 becomes `:crying_face:`.

Upsampling ratios are evaluated for each new round of training data with multipliers of 1, 5, 10, 100. For the R1–4 data, we carry forward the upsampling from Vidgen et al. (2021b): R1 is upsampled five times, R2 is upsampled 100 times, R3 is upsampled once, and R4 is upsampled once. Combining the model architectures and upsampling ratios gives 8 candidate models for each round’s target model. All models were implemented using the transformers library (Wolf et al., 2020). All models were trained for 3 epochs with early stopping based on the dev set loss, a learning rate of  $2e - 5$  and a weighted Adam optimizer. Other hyperparameters were set to HuggingFace defaults. We train and evaluate each model once and report results for this single run. Training took approximately 7 hours for each BERTweet model and 15 hours for each DeBERTa model using 8 NVIDIA Tesla V100 GPU on the JADE2 supercomputer.

The best target model for each round is selected by weighted accuracy between all prior rounds and the current round. For R6-T, a DeBERTa model with 100x upsampling on R5 performs best. Given the dearth of emoji in R1–R4, it is unsurprising a large upsample improves performance on emoji-based hate. For R7-T, DeBERTa with one upsampling on R6 performs best. For R8-T, DeBERTa with five upsamples on R7 performs best. In all rounds, DeBERTa significantly outperforms BERTweet, while the upsampling ratio less substantially affects performance. This suggests first converting emoji to their text representations does not substantially aid performance, though there may be other differences between the models driving results. We upload models to Dynalab for model-in-the-loop evaluation and data collection (Ma et al., 2021).

<sup>12</sup>This survey is on our [Github](#).



Table 8: Performance of models on emoji, text and all adversarial test sets, alongside benchmark evaluation sets: HATEMOJICHECK and HATECHECK.

	Emoji Test Sets				Text Test Sets				All Rounds	
	R5-R7		HMOJICHECK		R1-R4		HATECHECK		R1-R7	
	<i>n</i> = 593		<i>n</i> = 3930		<i>n</i> = 4119		<i>n</i> = 3728		<i>n</i> = 4712	
	Acc	F1	Acc	F1	Acc	F1	Acc	F1	Acc	F1
<b>P-IA</b>	0.508	0.394	0.689	0.754	0.679	0.720	0.765	0.839	0.658	0.689
<b>P-TX</b>	0.523	0.448	0.650	0.711	0.602	0.659	0.720	0.813	0.592	0.639
<b>B-D</b>	0.489	0.270	0.578	0.636	0.589	0.607	0.632	0.738	0.591	0.586
<b>B-F</b>	0.496	0.322	0.552	0.605	0.562	0.562	0.602	0.694	0.557	0.532
<b>R5-T</b>	0.585	0.490	0.779	0.825	<b>0.828</b>	<b>0.847</b>	0.956	0.968	0.786	0.801
<b>R6-T</b>	0.757	<b>0.769</b>	<b>0.879</b>	<b>0.910</b>	0.823	0.837	0.961	0.971	0.813	0.825
<b>R7-T</b>	<b>0.759</b>	0.762	0.867	0.899	0.824	0.842	0.955	0.967	0.813	0.829
<b>R8-T</b>	0.744	0.755	0.871	0.904	0.827	0.844	<b>0.966</b>	<b>0.975</b>	<b>0.814</b>	<b>0.829</b>

## G Robustness Analysis of Baselines

In addition to the models analyzed in the main paper, we evaluate three further models. The first is the ‘toxicity’ attribute returned by Perspective API (P-TX). We test this model because toxicity ratings are the most popular attributes.<sup>13</sup> The second and third models are two uncased BERT models (Devlin et al., 2019) trained on publicly-available academic datasets. B-D is trained on the Davidson et al. (2017) dataset of 24,783 tweets, labeled as *hateful*, *offensive* and *neither*. B-F is trained on the Founta et al. (2018) dataset of 99,996 tweets, labeled as *hateful*, *abusive*, *spam* and *normal*. Any labels besides *hateful* are binarized into a single *non-hateful* label. Two factors motivated the testing of these models. (1) There is a lack of emoji in the training data: the Davidson et al. dataset has 5.8% hateful cases, but only 7.4% of these contain emoji, and the Founta et al. dataset has 5.0% hateful cases, of which 14.7% contain emoji. (2) Despite BERT being a commonly-used architecture for hate speech detection, it encodes emoji as <UNK> tokens by default. We compare performance of the full set of pre-emoji models including those analyzed in the main paper (P-IA, P-TX, B-D, B-F, R5-T) versus our ‘emoji-aware’ models from each round of data collection (R6-T, R7-T, R8-T). Tab. 8 shows performance against the adversarially produced datasets in the emoji rounds of HATEMOJIBUILD and text rounds from Vidgen et al. (2021b), alongside two benchmark evaluation sets HATEMOJICHECK and HATECHECK (Röttger et al., 2021). P-TX has comparable performance to P-IA. B-D and B-F perform poorly on HATEMOJICHECK (F1 = 0.64, 0.61), and even more poorly on the adversarial test sets of emoji (F1 = 0.27, 0.32).

<sup>13</sup><https://support.perspectiveapi.com/s/about-the-api-faqs>

## H Fairness Considerations

As well as the six metrics discussed in the main paper, Tab. 9 shows two further fairness metrics: (1) the demographic parity ratio (Agarwal et al., 2019), which is equal to one when the selection rate is balanced across subgroups, and (2) the equalized odds ratio (Hardt et al., 2016; Agarwal et al., 2018), which is equal to one when the true positive, true negative, false positive, and false negative rates are balanced across subgroups. The academic BERT models perform worst (0.14 to 0.28 across the metrics), followed by the Perspective models (0.41 to 0.67). The most fair model is R5-T with Demographic Parity Ratio of 0.90 and Equalized Odds Ratio of 0.77. However, the overall performance of the emoji-aware models is higher (see Fig. 2): for every subgroup, R8-T has higher accuracy, precision and recall, and lower false positive and negative rates. Thus, the marginally worse performance by between-group fairness metrics is paired with a better ability to protect all these subgroups against emoji-based hate.

Table 9: Between-group fairness metrics by model.

	Demographic Parity Ratio	Equalized Odds Ratio
<b>P-IA</b>	0.668	0.430
<b>P-TX</b>	0.611	0.405
<b>B-D</b>	0.174	0.138
<b>B-F</b>	0.276	0.257
<b>R5-T</b>	<b>0.898</b>	<b>0.767</b>
<b>R6-T</b>	0.818	0.711
<b>R7-T</b>	0.848	0.667
<b>R8-T</b>	0.832	0.600