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

Hannah Rose Kirk[1‡], **Bertie Vidgen**[1], **Paul Röttger**[1], **Tristan Thrush**[2], **Scott A. Hale**[1,3]
[1]University of Oxford, [2]Facebook AI Research, [3]Meedan
[‡]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 a key 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 HATEMOJITRAIN dataset using a human-and-model-in-the-loop approach. Models trained on these 5,912 adversarial examples perform substantially better at detecting emoji-based hate, while retaining strong performance on text-only hate. Both HATEMOJICHECK and HATEMOJITRAIN are made publicly available.

1 Introduction

Online hate harms its targets, disrupts online communities, pollutes civic discourse and reinforces social power imbalances (Gelber, 2017). The sheer scale of hateful content online has led to widespread use of automated detection systems to find, monitor and stop it (Waseem et al., 2018; Vidgen et al., 2019; Gillespie, 2020; Caselli et al., 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 which 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 over 10 billion 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:

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

2. **RQ2**: To what extent does 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 the 'Identity Attack' model from 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 trick the model-in-the-loop. This process yields a dataset of 5,912 entries, half of which are challenging contrasts. We call this dataset HATEMOJITRAIN. 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 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 HATEMOJITRAIN, 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 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 HATEMOJITRAIN publicly available.[1]

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).[2]

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

2 HATEMOJICHECK: Functional Tests for Emoji-Based Hate

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 selected the functionalities in HATEMOJICHECK to be (1) realistic: functionalities capture real-world uses of emoji-based hate, (2) unique: each functionality covers a distinct aspect of emoji-based hate, without overlaps between functionalities, and (3) unambiguous: functionalities have clear gold-standard labels. The functionalities were motivated from two perspectives.

Previous Literature We identified distinct uses of emoji in online communications, particularly online 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 reviewed a wider range of literature on online hate, identifying distinct types of hateful language (Waseem and Hovy, 2016; Zampieri et al., 2019; Vidgen et al., 2021a).

Empirical Investigation We defined a list of hateful emoji and words, and then used Twitter's Streaming API to search for the Cartesian products of emoji–emoji and emoji–word pairs from this list over two weeks in April 2021. To identify different forms of emoji-based hate, we applied a grounded theory approach on a sample of 3,295 tweets (Corbin and Strauss, 1990), splitting out distinctive categories, and then recursively selecting sub-categories until all key parts of the data were captured and the framework was 'saturated'.

2.2 Functionalities in HATEMOJICHECK

HATEMOJICHECK has seven functionalities, as shown in 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 (🕌). **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 used for Black people: (🦍, 🍌) and Muslims: (🐐, 🐪). For expressions of disgust, three descriptors are used: 🤮, 😷, 🤢. **F4: Double Swap** tests a combination of **F1** with both **F2** and **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 😷".

[1] The HATEMOJICHECK and HATEMOJITRAIN datasets were generated, tested, and released by the University of Oxford. Available at https://github.com/HannahKirk/Hatemoji.

[2] We recognize the presence of annotator bias in cultural interpretations of this definition, and provide data statements for both datasets in the appendix.

Table 1: Summary of the seven functionalities in HATEMOJICHECK.

Functionality	Example From Twitter	Templates	Emoji	Identities	Cases
***F1** Verb Swap*: Swap threatening verb	"I will 🔪 Muslims"	10	5	6	300
***F2** Identity Swap*: Swap protected identity	"I hate 🧔"	20	1	6	120
***F3** Descriptor Swap*: Swap descriptor noun/emotion	"Muslims are like 🐀"	10	3–7	6	260
***F4** Double Swap*: Combine two of the above swaps	"Please ♿🔫"	10	3–7	6	288
***F5** Append*: Append neg. emoji to neutral text	"Imagine being a muslim 😭"	10	3–5	6	288
***F6** Positive Confounder*: Append pos. emoji to hateful text	"kill the gays 😇"	13	5	6	440
***F7** Emoji Leetspeak*: Substitute for character or word-piece	"I hate n1ggers I really do"	15	3-4	6 (+16 slurs)	430
				Total Test Cases	2126

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 Muslim" is neutral but "My new manager is Muslim 😭" 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

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 (gender), trans people (gender identity), gay people (sexual orientation), Black people (race), disabled people (disability) and Muslims (religion). For templates that contain slurs, we use the 16 slurs compiled by Röttger et al. (2021) (see Appendix B.1). Second, each template is given an emoji placeholder which is populated by emoji that share semantic and syntactic properties, so that the test suite can be scaled across different emoji. For example, we use the template "I want to [EMOJI] all [IDENTITY]" to generate the test case "I want to 🔪 all Muslims". HATEMOJICHECK comprises 3,930 entries. There are 2,126 original test cases and 1,804 perturbations (see Tab. 1 for detail).

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.[3] Test cases generated from these perturbed templates share vocabulary and syntax with the original test cases, enabling fine-grained assessment of model strengths and 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 identities (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 perturbations 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

[3] See Appendix B.2 for details on perturbations.

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** 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.[4] Annotators were given extensive guidelines, 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 core 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 Training Better Models with HATEMOJITRAIN

As reported in Sec. 4, we find existing models perform poorly on emoji-based hate as measured with HATEMOJICHECK. We address those failings by 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.[5]

Dataset Generation From 24 May to 11 June 2021 we implemented three successive rounds of data generation and model re-training to create the HATEMOJITRAIN dataset. In each round we tasked a team of 10 trained annotators with entering content which would trick the model-in-the-loop, which we refer to as the target model.[6] 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, then hateful content was assigned secondary labels for the type and target of hate. This is similar to the approach used in Zampieri et al. (2019) and Vidgen et al. (2021b,a). Each entry was validated by two additional annotators and an expert annotator resolved disagreements. Annotators then created a perturbation for each entry. To maximize sentence similarity between originals and perturbations, annotators could either make an emoji substitution while holding the text fixed, or fix the emoji and minimally change the surrounding text. Each perturbation received two additional annotations, and disagreements were resolved by the expert annotator. This weekly cadence of annotator tasks was repeated in three consecutive weeks. The dataset composition is described in Appendix D.

Model Implementation For our first round of data generation (R5) the target model is **R5-T**, the DeBERTa model released by Ma et al. (2021), which was trained on the dynamically-generated dataset from Vidgen et al. (2021b) as well as 468,928 entries compiled from 11 English-language hate speech datasets. This model has been shown to perform well on text-only hate, but has seen limited emoji in training. At the end of each round, the data for that round is assigned a 80:10:10 train/dev/test split. The train split is then upsampled to improve performance with increments of $1, 5, 10, 100$, with the optimum upsampling taken forward to all subsequent rounds. The target model is then re-trained on the training data from all prior rounds as well as the current round.[7] We evaluate models and upsampling ratios at the end of each round by weighted accuracy with 50% weight on the test sets from all prior rounds and 50% weight on current round test set. We use weighted accuracy so that we can assess performance against the latest (emoji-specific and most-adversarial) round without risking overfitting and reducing performance on the previous test sets.

[4] See annotator demographics in Appendix A.

[5] Dynabench is an open-source platform which supports dynamic dataset generation and model benchmarking for a variety of NLP tasks. See: https://dynabench.org/

[6] See annotator demographics in Appendix C.

[7] For further details of model training, see Appendix E.

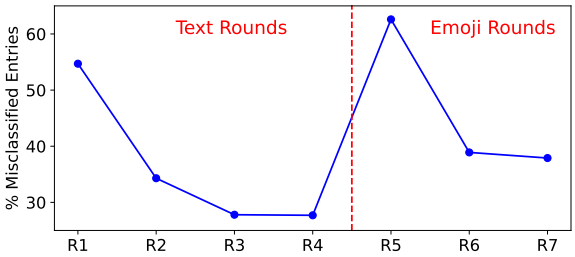

Figure 1: MER for R1–4 (text) and R5–R7 (emoji).

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 (Vidgen et al., 2021b). 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 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 R1–4 test set, with an F1 of 0.847, but only scores 0.490 on the newly generated R5–7 test sets from HATEMOJITRAIN. **R6-T**, **R7-T** and **R8-T** perform better on R5–7, with F1 between 0.744 and 0.759. They perform similarly well on the R1–4 test set, with F1 from 0.837 to 0.844, 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 0.829. The greatest performance gain in the adversarially trained models is from **R5-T** to **R6-T**, with an increase in F1 of 0.28. This is achieved with only 2,000 emoji-containing examples. By comparison, **R7-T** and **R8-T** yield very small improvements.

4 Evaluating Models with HATEMOJICHECK

We present results for two existing models as baselines, with additional baselines shown in Appendix F. The first baseline is Google Jigsaw's Perspective API, a widely-used commercial tool for content moderation.[8] 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 perturbations) 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.

Table 2: Performance of target models on emoji, text and all adversarially-collected test sets.

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

On HATEMOJICHECK as a whole, our newly trained models **R6-T**, **R7-T** and **R8-T** perform best, with overall accuracy from 0.867 to 0.879 (Tab. 3). They substantially outperform **R5-T**, with accuracy of 0.779, and **P-IA**, accuracy of 0.689. Our newly trained models have the smallest emoji difference, between 0.033 and 0.075. In contrast, **P-IA** has an emoji difference of 0.159 and **R5-T** of 0.217. Comparing the three models trained on HATEMOJITRAIN, the first round of adversarial data 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 HATEMOJITRAIN 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**, **R5-T** and **P-IA** perform well on original statements but then perform poorly on the Polarity perturbations (0.42 and 0.20 accuracy, respectively). Our models have much stronger

[8] https://www.perspectiveapi.com/

Table 3: Aggregate accuracy for models evaluated on HATEMOJICHECK.

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

performance on Polarity perturbations (between 0.73 and 0.77), and comparable high performance on the other sets of test cases. For **F2 Identity Swap**, **R5-T** and **P-IA** perform very poorly on original statements (0.33 and 0.14 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 **R5-T** and **P-IA** 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**, **R5-T** and **P-IA** perform moderately with accuracies of 0.69 and 0.73. 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 **R5-T** and **P-IA** 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**. Prior solutions perform well on original statements and poorly on perturbations precisely because they ignore the effect of the emoji confounder. **F7 Emoji Leetspeek** 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 HATEMOJITRAIN perform well on emoji-based hate, but there is no clear 'best' model among them. After the large improvement from **R6-T** over **R5-T**, subsequent target models produce minimal performance gains, or in some cases minimal performance losses, across the functionalities.

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. This indicates that the models do not understand what the identity emoji represent. Existing solutions 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 solutions, 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 more rounds of 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 HATEMOJITRAIN. This suggests that while training on a relatively small number of entries can substantially improve performance,

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

	Functionality	Label	Example Test Case Template	n	Pre-Emoji P-IA	Pre-Emoji R5-T	Post-Emoji R6-T	Post-Emoji R7-T	Post-Emoji R8-T
F1	**Verb Swap**	**H**	I want to 🔪 all [IDENTITY]	300	0.94	0.85	0.84	0.79	0.84
F1.1	Identity Perturb	**NH**	I want to 🔪 all bigots	50	0.94	1.00	1.00	0.98	0.86
F1.2	Polarity Perturb	**NH**	I want to 🙏 all [IDENTITY]	60	0.20	0.42	0.75	0.77	0.73
F1.3	No Emoji Perturb	**H**	I want to kill all [IDENTITY]	60	1.00	0.95	0.97	0.97	0.98
				emoji diff:	-0.06	-0.10	-0.12	-0.18	-0.15
F2	**Identity Swap**	**H**	I hate [IDENTITY EMOJI]	120	0.14	0.33	0.83	0.83	0.83
F2.1	Identity Perturb	**NH**	I hate 🍕	20	0.90	0.70	1.00	0.85	1.00
F2.2	Polarity Perturb	**NH**	I love [IDENTITY EMOJI]	120	1.00	0.98	0.78	0.88	0.93
F2.3	No Emoji Perturb	**H**	I hate [IDENTITY]	120	0.98	0.98	0.97	1.00	0.95
				emoji diff:	-0.84	-0.66	-0.13	-0.17	-0.12
F3	**Descriptor Swap**	**H**	[IDENTITY] are nothing but 🐀	260	0.92	0.83	0.99	0.99	1.00
F3.1	Identity Perturb	**NH**	Bullies are nothing but 🐀	40	1.00	1.00	1.00	1.00	1.00
F3.2	Polarity Perturb	**NH**	[IDENTITY] are nothing but ❤️	60	0.25	0.48	0.78	0.82	0.93
F3.3	No Emoji Perturb	**H**	[IDENTITY] are nothing but vermin	60	0.98	1.00	1.00	1.00	1.00
				emoji diff:	-0.06	-0.17	-0.01	-0.01	0.00
F4	**Double Swap**	**H**	I will 🔫 [IDENTITY EMOJI]	288	0.00	0.03	0.79	0.70	0.77
F4.1	Identity Perturb	**NH**	I will 🔫 🕷	46	1.00	1.00	0.91	0.98	0.91
F4.2	Polarity Perturb	**NH**	I will ❤️ [IDENTITY EMOJI]	60	1.00	0.98	0.92	0.85	0.92
F4.3	No Emoji Perturb	**H**	I will hurt [IDENTITY]	60	0.97	1.00	1.00	1.00	1.00
				emoji diff:	-0.96	-0.97	-0.21	-0.30	-0.23
F5	**Append**	**H**	My new manager is [IDENTITY] 😭	288	0.73	0.69	0.99	0.89	0.87
F5.1	Identity Perturb	**NH**	My new manager is a cat-lover 😭	48	1.00	1.00	1.00	1.00	1.00
F5.2	Polarity Perturb	**NH**	My new manager is [IDENTITY] 👍	60	0.47	0.55	0.85	0.78	0.75
F5.3	No Emoji Perturb	**NH**	My new manager is [IDENTITY]	60	0.35	0.48	0.45	0.43	0.37
				emoji diff:	NA	NA	NA	NA	NA
F6	**Positive Confounder**	**H**	I find all [IDENTITY] disgusting 🙌	440	0.93	1.00	0.85	0.90	0.89
F6.1	Identity Perturb	**NH**	I find all reptiles disgusting 🙌	65	0.75	0.92	0.92	0.92	0.92
F6.2	Polarity Perturb	**NH**	I find all [IDENTITY] inspiring 🙌	112	0.44	0.88	0.96	0.91	0.93
F6.3	No Emoji Perturb	**H**	I find all [IDENTITY] disgusting	88	0.93	0.99	0.95	1.00	0.93
				emoji diff:	0.00	0.01	-0.10	-0.10	-0.04
F7	**Emoji Leetspeak**	**H**	I hate [SLUR_EMOJI]	430	0.59	0.85	0.91	0.83	0.91
F7.1	Identity Perturb	**NH**	I hate d0gs	45	0.82	0.67	0.62	0.64	0.49
F7.2	Polarity Perturb	**NH**	I hate people who say [SLUR_EMOJI]	430	0.57	0.79	0.79	0.86	0.77
F7.3	No Emoji Perturb	**H**	I hate [SLUR]	140	0.61	0.99	0.99	0.94	0.99
				emoji diff:	-0.03	-0.14	-0.09	-0.11	-0.07

the gains quickly saturate. Future work could investigate performance differences in more detail using the labels for type and target of hate which 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 offers negative predictive power: high performance only indicates the absence of weaknesses (Gardner et al., 2020). Second, it only includes relatively simplistic statements and a constrained set of emoji. Future work could address evaluation 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. These limitations do not diminish HATEMOJICHECK's utility as a tool for effectively identifying model weaknesses around emoji-based hate. By publicly releasing the test suite, we encourage practitioners and academics to scrutinize their models prior to deployment or publication.

Adversarial generation of data 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 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. In studies which do attend to emoji, they 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 which 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 an iterative human-and-model-in-the-loop training system 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., 2020). Earlier work by Dinan et al. (2020) 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, which covers seven functionalities with 3,930 test cases. Using this test suite, we exposed clear weaknesses in the performance of a commercial model and existing academic models on emoji-based hate. To address these weaknesses, we created the HATEMOJITRAIN dataset using an innovative human-and-model-in-the-loop approach. We showed that models trained on this adversarial data perform substantially better at detecting emoji-based hate, while retaining strong performance on text-only hate. Our approach of first identifying granular model weaknesses then creating a targeted training dataset to address them presents a promising direction for better detecting other diverse and emerging forms of online harm.

Acknowledgments

We are thankful for funding that the Oxford authors received to support annotation from the Volkswagen Foundation, Meedan, Keble College, the Oxford Internet Institute, and Rewire Online Ltd. We owe a debt of gratitude to all our annotators and Dynabench. 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

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.

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.

Brandwatch. 2018. The Emoji Report.

Spencer Cappallo, Stacey Svetlichnaya, Pierre Garrigues, Thomas Mensink, and Cees G. M. Snoek. 2018. The New Modality: Emoji Challenges in Prediction, Anticipation, and Retrieval. *IEEE Transactions on Multimedia*, 21(2):402–415.

Tommaso Caselli, Valerio Basile, Jelena Mitrović, Inga Kartoziya, and Michael Granitzer. 2020. I Feel Offended, Don't Be Abusive! Implicit/Explicit Messages in Offensive and Abusive Language. In *Proceedings of the 12th Language Resources and Evaluation Conference*, pages 6193–6202.

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, Stroudsburg, PA, USA. Association for Computational Linguistics.

Juliet M. Corbin and Anselm Strauss. 1990. Grounded theory research: Procedures, canons, and evaluative criteria. *Qualitative Sociology 1990 13:1*, 13(1):3–21.

Thomas Davidson, Dana Warmsley, Michael Macy, and Ingmar Weber. 2017. Automated Hate Speech Detection and the Problem of Offensive Language. *Proceedings of the 11th International Conference on Web and Social Media, ICWSM 2017*, pages 512–515.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. *NAACL HLT 2019 - 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies - Proceedings of the Conference*, 1:4171–4186.

Emily Dinan, Samuel Humeau, Jason Weston, and Bharath Chintagunta. 2020. Build it break it fix it for dialogue safety: Robustness from adversarial human attack. In *EMNLP-IJCNLP 2019 - 2019 Conference on Empirical Methods in Natural Language Processing and 9th International Joint Conference on Natural Language Processing, Proceedings of the Conference*, pages 4537–4546. Association for Computational Linguistics.

Lucas Dixon, John Li, Jeffrey Sorensen, Nithum Thain, and Lucy Vasserman. 2018. Measuring and Mitigating Unintended Bias in Text Classification. *AIES 2018 - Proceedings of the 2018 AAAI/ACM Conference on AI, Ethics, and Society*, 7(18):67–73.

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. *EMNLP 2017 - Conference on Empirical Methods in Natural Language Processing, Proceedings*, pages 1615–1625.

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. *12th International AAAI Conference on Web and Social Media, ICWSM 2018*, pages 491–500.

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 Khashabi, 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, Stroudsburg, PA, USA. Association for Computational Linguistics.

Katharine Gelber. 2017. Hate Speech–Definitions & Empirical Evidence. *Constitutional Commentary*, 559.

Tarleton Gillespie. 2020. Content moderation, AI, and the question of scale. *Big Data and Society*, 7(2).

Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. 2020. DeBERTa: Decoding-enhanced BERT with Disentangled Attention.

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*, pages 1–5, New York, NY, USA. ACM.

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.

J. Richard Landis and Gary G. Koch. 1977. The Measurement of Observer Agreement for Categorical Data. *Biometrics*, 33(1):159.

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.

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. *CoRR*, abs/2106.06052.

Dat Quoc Nguyen, Thanh Vu, and Anh Tuan Nguyen. 2020. BERTweet: A pre-trained language model for english tweets.

Christopher Potts, Zhengxuan Wu, Atticus Geiger, and Douwe Kiela. 2020. DynaSent: A Dynamic Benchmark for Sentiment Analysis. pages 2388–2404.

J. J Randolph. 2005. Free-Marginal Multirater Kappa (multirater K [free]): An Alternative to Fleiss. In *Joensuu Learning and Instruction Symposium*, October, 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, Stroudsburg, PA, USA. 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, Bertram Vidgen, Dong Nguyen, Zeerak Waseem, Helen Margetts, and Janet B. Pierrehumbert. 2021. HateCheck: Functional Tests for Hate Speech Detection Models. *ACL 2021 - 59th Annual Meeting of the Association for Computational Linguistics, Proceedings of the Conference*.

Niloofar Safi Samghabadi, Afsheen Hatami, Mahsa Shafaei, Sudipta Kar, and Thamar 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.

Trieu H. Trinh and Quoc V. Le. 2018. A Simple Method for Commonsense Reasoning.

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, Stroudsburg, PA, USA. 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 *ACL 2021 - 59th Annual Meeting of the Association for Computational Linguistics, Proceedings of the Conference*.

Zeerak Waseem and Dirk Hovy. 2016. Hateful Symbols or Hateful People? Predictive Features for Hate Speech Detection on Twitter. In *Proceedings of the NAACL Student Research Workshop*, pages 88–93, Stroudsburg, PA, USA. Association for Computational Linguistics.

Zeerak Waseem, James Thorne, and Joachim Bingel. 2018. Bridging the Gaps: Multi Task Learning for Domain Transfer of Hate Speech Detection. In *Online Harassment*, pages 29–55. Springer, Cham.

Michael Wiegand and Josef Ruppenhofer. 2021. Exploiting Emojis for Abusive Language Detection. Technical report.

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. 2019. HuggingFace's Transformers: State-of-the-art Natural Language Processing.

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, Stroudsburg, PA, 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. Technical report.

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.

A Data Statement for HATEMOJICHECK

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

A.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 (*hateful* versus *not hateful*). 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.

A.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.

A.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.

A.4 Annotator Demographics

We recruited a team of three annotators who worked for two weeks in May 2021. 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.

A.5 Speech Situation

The modality of all test cases is written text embedded with emoji. 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.

A.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. 67.5% 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.

B Constructing HATEMOJICHECK

B.1 List of Hateful Slurs in HATEMOJICHECK

The slurs used in HATEMOJICHECK are selected in the same method as for HATECHECK. For each of the six protected identities in HATEMOJICHECK, the most common slurs are taken from `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 third ranked slur is significantly less common than the second ranked. Each identity thus has two or three associated slurs (see Tab. 5).

B.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 template we use the same emoji for

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*pple, m*ng
Muslims	m*zzie, J*hadi, camelf*cker

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 identity (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 of 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.

C Data Statement for HATEMOJITRAIN

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

C.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 inputed 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.

C.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 be could be adapted for other languages in future work.

C.3 Speaker Demographics

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

C.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, 73% were between 18–29 years old and 27% between 30–39 years old. The completed education level was high school for 27% of annotators, undergraduate degree for 9% of annotators, taught graduate degree for 36% of annotators and postgraduate research degree for 27% of annotators. 54.5% of annotators were female, and 45.5% were male. Annotators came from a variety of nationalities, with 64% British, as well as Jordanian, Irish, Polish and Spanish. 64% of annotators identified as ethnically White and the remaining annotators came from various ethnicities including Turkish, Middle Eastern, and Mixed White and South Asian. 36% of annotators were Muslim, and others identified as Atheist or as having no religious affiliation. 82% of annotators were native English speakers and 18% were non-native but fluent. The majority of annotators (82%) used emoji and social media more than once per day. 91% of annotators had seen others targeted by abuse online, and 36% had been personally targeted.

C.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.

C.6 Text Characteristics

The genre of texts is hateful and non-hateful statements using emoji constructions. Annotators in-

putted emoji into the platform using a custom emoji picker (https://hatemoji.stackblitz.io/). 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.

D Constructing HATEMOJITRAIN

Types of Hate We adopt the same categorization used by Vidgen et al. (2021b, p.3).[9] There are four types of hate. Derogation: Language which explicitly derogates, demonizes, demeans or insults 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 Each round has approximately the same number of entries with slightly fewer in R7 due to more quality control issues (see Tab. 6). Labels are equally distributed in each round. 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-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 (Cappallo et al., 2018; Felbo et al., 2017; Bick, 2020).

[9] 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 HATEMOJITRAIN.

		R5	R6	R7
n		1994	1966	1952
Split, n (%)	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, n (%)	hate	1006 (50.5)	983 (50.0)	976 (50.0)
	not hate	988 (49.5)	983 (50.0)	976 (50.0)
Set, n (%)	original	997 (50.0)	983 (50.0)	976 (50.0)
	perturbation	997 (50.0)	983 (50.0)	976 (50.0)
Type, n (%)	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, μ (σ)		1.7 (2.2)	1.7 (1.0)	1.6 (1.1)

E Target Models

In each round we assessed two candidate model architectures. The first is an uncased DeBERTa base model with a sequence classification head (He et al., 2020). DeBERTa has been shown to improve on BERT (Devlin et al., 2018) 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 CommonCrawl (Trinh and Le, 2018). The second candidate is an uncased BERTweet model with a sequence classification head where the RoBERTa training procedure was repeated on 850M English Tweets using a custom vocabulary, which may mean it is more 'emoji-aware' (Nguyen et al., 2020).

Upsampling ratios are evaluated for each new round of training data with increments 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,

Table 7: 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 $n = 593$		HMOJICHECK $n = 3930$		R1-R4 $n = 4119$		HATECHECK $n = 3728$		R1-R7 $n = 4712$	
	Acc	**F1**	**Acc**	**F1**	**Acc**	**F1**	**Acc**	**F1**	**Acc**	**F1**
P-IA	0.508	0.394	0.689	0.754	0.679	0.720	0.765	0.839	0.658	0.689
P-TX	0.523	0.448	0.650	0.711	0.602	0.659	0.720	0.813	0.592	0.639
B-D	0.489	0.270	0.578	0.636	0.589	0.607	0.632	0.738	0.591	0.586
B-F	0.496	0.322	0.552	0.605	0.562	0.562	0.602	0.694	0.557	0.532
R5-T	0.585	0.490	0.779	0.825	**0.828**	**0.847**	0.956	0.968	0.786	0.801
R6-T	0.757	**0.769**	**0.879**	**0.910**	0.823	0.837	0.961	0.971	0.813	0.825
R7-T	**0.759**	0.762	0.867	0.899	0.824	0.842	0.955	0.967	0.813	0.829
R8-T	0.744	0.755	0.871	0.904	0.827	0.844	**0.966**	**0.975**	**0.814**	**0.829**

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., 2019). 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. Training took approximately 7 hours for each BERTweet model and 15 hours for each DeBERTa model using 8 GPUs 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**, a DeBERTa model with one upsample on R6 performs best. For **R8-T**, a DeBERTa model with five upsamples on R7 performs best. In all rounds, DeBERTa significantly outperforms BERTweet, while the upsampling ratio less substantially affects performance. We use Dynalab to upload models for model-in-the-loop evaluation and data collection (Ma et al., 2021).

F 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.[10] The second and third model are two uncased BERT models (Devlin et al., 2018) 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. 7 shows performance against the adversarially produced datasets in the emoji rounds of HATEMOJITRAIN and text rounds from Vidgen et al. (2021b), alongside two benchmark evaluation sets HATEMOJICHECK and HATECHECK. **P-TX** has comparable performance to **P-IA**. **B-D** and **B-F** perform poorly on HATEMOJICHECK (F1 = 0.636, 0.605), and even more poorly on the adversarial test sets of emoji (F1 = 0.270, 0.322).

[10] https://support.perspectiveapi.com/s/about-the-api-faqs