Hypothesis Engineering for Zero-Shot Hate Speech Detection

Janis Goldzycher and **Gerold Schneider**
Department of Computational Linguistics
University of Zurich
{goldzycher,gschneid}@cl.uzh.ch

Abstract

Standard approaches to hate speech detection rely on sufficient available hate speech annotations. Extending previous work that repurposes natural language inference (NLI) models for zero-shot text classification, we propose a simple approach that combines multiple hypotheses to improve English NLI-based zero-shot hate speech detection. We first conduct an error analysis for vanilla NLI-based zero-shot hate speech detection and then develop four strategies based on this analysis. The strategies use multiple hypotheses to predict various aspects of an input text and combine these predictions into a final verdict. We find that the zero-shot baseline used for the initial error analysis already outperforms commercial systems and fine-tuned BERT-based hate speech detection models on HateCheck. The combination of the proposed strategies further increases the zero-shot accuracy of 79.4% on HateCheck by 7.9 percentage points (pp), and the accuracy of 69.6% on ETHOS by 10.0pp.[1]

1 Introduction

With the increasing popularity of social media and online forums, phenomena such as hate speech, offensive and abusive language, and personal attacks have gained a powerful medium through which they can propagate fast. Due to the sheer number of posts and comments on social media, manual content moderation has become unfeasible, thus the automatic detection of harmful content becomes essential. In natural language processing, there now exist established tasks with the goal of detecting offensive language (Pradhan et al., 2020), abusive language (Nakov et al., 2021), hate speech (Fortuna and Nunes, 2018) and other related types of harmful content (Poletto et al., 2021). In this work, we focus on the detection of hate speech, which is typically defined as attacking, abusive or discriminatory language that targets people on the basis of identity defining group characteristics such as gender, sexual orientation, disability, race, religion, national origin etc. (Fortuna and Nunes, 2018; Poletto et al., 2021; Yin and Zubiaga, 2021). Most current hate speech detection approaches rely on either training models from scratch or fine-tuning pre-trained language models (Jahan and Oussalah, 2021). Both types of approaches need large amounts of labeled data which are only available for a few high-resource languages (Poletto et al., 2021) and costly to create. Therefore, exploring data-efficient methods for hate speech detection is an attractive alternative.

In this paper, we build on Yin et al. (2019) who proposed to re-frame text classification tasks as natural language inference, enabling high accuracy zero-shot classification. We exploit the fact that we can create arbitrary hypotheses to predict aspects of an input text that might be relevant for hate speech detection. To identify effective hypotheses, we first find a well-performing hypothesis formulation that claims that the input text contains hate speech. An error analysis based on HateCheck (Röttger et al., 2021) shows that given a well-performing formulation the model still struggles with multiple phenomena, including (1) abusive or profane language that does not target people based on identity-defining group characteristics, (2) counterspeech, (3) reclaimed slurs, and (4) implicit hate speech. To mitigate these misclassifications, we develop four strategies. Each strategy consists of multiple hypotheses and rules that combine these hypotheses in order to address one of the four identified error types.

We show that the combination of all proposed strategies improves the accuracy of vanilla NLI-based zero-shot prediction by 7.9pp on HateCheck (Röttger et al., 2021) and 10.0pp on ETHOS (Mollas et al., 2022). An error analysis shows that

[1] The code and instructions to reproduce the experiments are available at https://github.com/jagol/nli-for-hate-speech-detection.

Proceedings of the 29th International Conference on Computational Linguistic, pages 75–90
October 12–17, 2022.

the overall gains in accuracy largely stem from increased performance on previously identified weaknesses, demonstrating that the strategies work as intended.

Overall, our primary contributions are the following:

C1 An error analysis of vanilla NLI-based zero-shot hate speech detection.

C2 Developing strategies that combine multiple hypotheses to improve zero-shot hate speech detection.

C3 An evaluation and error analysis of the proposed strategies.

2 Background and Related Work

Early approaches to hate speech detection have focused on English social media posts, especially Twitter, and treated the task as binary or ternary text classification (Waseem and Hovy, 2016; Davidson et al., 2017; Founta et al., 2018). In more recent work, additional labels have been introduced that indicate whether the post is group-directed or not, who the targeted group is, if the post calls for violence, is aggressive, contains stereotypes, if the hate is expressed implicitly, or if sarcasm or irony is used (Mandl et al., 2019, 2020; Sap et al., 2020; ElSherief et al., 2021; Röttger et al., 2021; Mollas et al., 2022). Sometimes hate speech is not directly annotated but instead labels, such as *racism*, *sexism*, *homophobia* that already combine hostility with a specific target are annotated and predicted (Waseem and Hovy, 2016; Waseem, 2016; Saha et al., 2018; Lavergne et al., 2020).

While early approaches relied on manual feature engineering (Waseem and Hovy, 2016), most current approaches are based on pre-trained transformer-based language models that are then fine-tuned on hate speech datasets (Florio et al., 2020; Uzan and HaCohen-Kerner, 2021; Banerjee et al., 2021; Lavergne et al., 2020; Das et al., 2021; Nghiem and Morstatter, 2021).

Some work has focused on reducing the need for labeled data by multi-task learning on different sets of hate speech labels (Kapil and Ekbal, 2020; Safi Samghabadi et al., 2020) or adding sentiment analysis as an auxiliary task (Plaza-Del-Arco et al., 2021). Others have worked on reducing the need for non-English annotations by adapting hate speech detection models from high- to low-resource languages in a cross-lingual zero-shot setting (Stappen et al., 2020; Pamungkas et al., 2021). However the approach has been criticized for being unreliable when encountering language-specific taboo interjections (Nozza, 2021).

name	# examples	classes
HateCheck	3,728	hateful (68.8%), non-hate (31.2%)
ETHOS (binary)	997	hate speech (64.1%), not-hate speech (25.9%)

Table 1: The number of examples and the class balance of the datasets.

2.1 Zero-Shot Text Classification

The advent of large language models has enabled zero-shot and few-shot text classification approaches such as prompting (Liu et al., 2021), and task descriptions (Raffel et al., 2020), which convert the target task to the pre-training objective and are usually only used in combination with large language models. Chiu and Alexander (2021) use the prompts *“Is this text racist?”* and *“Is this text sexist?”* to detect hate speech with GPT-3. Schick et al. (2021) show that toxicity in large generative language models can be avoided by using similar prompts to self-diagnose toxicity during the decoding.

In contrast, NLI-based prediction in which a target task is converted to an NLI-task and fed into an NLI model converts the target task to the fine-tuning task. Here, a model is given a premise and a hypothesis and tasked to predict if the premise entails the hypothesis, contradicts it, or is neutral towards it. Yin et al. (2019) proposed to use an NLI model for zero-shot topic classification, by inputting the text to classify as the premise and constructing for each topic a hypothesis of the form “This text is about <topic>”. They map the labels *neutral* and *contradiction* to *not-entailment*. We can then interpret a prediction of entailment as predicting that the input text belongs to the topic in the given hypothesis. Conversely, *not-entailment* implies that the text is not about the topic. Wang et al. (2021) show for a range of tasks, including offensive language identification, that this task reformulation also benefits few-shot learning scenarios. Recently, AlKhamissi et al. (2022) obtained large performance improvements in few-shot learning for hate speech detection by (1) decomposing the task into four subtasks and (2) additionally training the few-shot model on a knowledge base.

3 Data

HateCheck Röttger et al. (2021) introduce this English, synthetic, evaluation-only dataset, annotated for a binary decision between hate speech and not-hate speech. It covers 29 functionalities that are either a type of hate speech or challenging types of non-hate speech that could be mistaken for hate speech by a classifier. The examples for each of these functionalities have been constructed on the basis of conversations with NGO workers. Each of these templates contains one blank space to be filled with a protected group. The authors fill these templates with seven protected groups, namely: women, gay people, transgender people, black people, Muslims, immigrants, and disabled people. Overall the dataset contains 3,728 examples.

ETHOS The ETHOS dataset (Mollas et al., 2022) is split into two parts: one part is annotated for the presence of hate speech. The other part contains fine-grained annotations that indicate which characteristics have been targeted (gender, sexual orientation, race, ethnicity, religion, national origin, disability), whether the utterance calls for violence, and whether it is directed at an individual or a general statement about a group. The dataset is based on English comments from Youtube and Reddit. For this work, we will only make use of the binary hate speech annotations. These annotations are continuous values between 0 (indicating no hate speech at all) and 1 indicating clear hate speech. We rounded all annotations to either 0 or 1 using a threshold of 0.5.

Table 1 displays the class balances of the two datasets.

4 Evaluating Standard Zero-Shot Prediction

The evaluation of standard zero-shot NLI-based hate speech detection has two goals: To (1) obtain an error analysis that serves as the starting point for developing zero-shot strategies in Section 5, and (2) establish a baseline for those strategies.

Experiment setup To test if an input text contains hate speech, we need a hypothesis expressing that claim. However, there are many ways how the claim, that a given text contains hate speech, can be expressed. Choosing a sub-optimal way to express this claim will result in lower accuracy. Wang et al. (2021) already tested four different hypotheses for hate speech or offensive language. We conduct an extensive evaluation by constructing and testing all grammatically correct sentences built with the following building blocks: *It/That/This* + *example/text* + *contains/is* + *hate speech/hateful/hateful content*. We conduct all experiments with a BART-large model (Lewis et al., 2020) that was fine-tuned on the **M**ulti-Genre **N**atural **L**anguage **I**nference dataset (MNLI) (Williams et al., 2018) and has been made available via the Huggingface transformers library (Wolf et al., 2020) as `bart-large-mnli`. This model predicts either *contradiction*, *neutral*, or *entailment*. We follow the recommendation of the model creators to ignore the logits for *neutral* and perform a softmax over the logits of *contradiction* and *entailment*. If the probability for entailment is equal or higher than 0.5 we consider this a prediction of *entailment* and thus *hate speech*.[3] We evaluate on HateCheck since the functionalities in this dataset allow for an automatic in-depth error analysis and compare our results to the baselines provided by Röttger et al. (2021).

system	acc. (%)
BART-MNLI 0-shot results	
That example is hate speech. / That is hateful.	66.6
That contains hate speech.	79.4
average	75.1
Systems evaluated by Röttger et al. (2021)	
SiftNinja	33.2
BERT fine-tuned on Davidson et al. (2017)	60.2
BERT fine-tuned on Founta et al. (2018)	63.2
Google Jigsaw Perspective [2]	76.6

Table 2: Evaluation of hypotheses for zero-shot hate speech detection on HateCheck. The top rows contain the two lowest scoring hypotheses, the highest scoring hypothesis and the average score for all tested hypotheses. The bottom rows contain the HateCheck baselines computed by Röttger et al. (2021). The full results for all tested hypotheses are listed in Appendix A.

Results Table 2 shows an abbreviated version of the results. The full results are given in Appendix A. The hypothesis “That contains hate speech.” obtains the highest accuracy and beats the Google-Jigsaw API by 2.8pp. This is remarkable, since we can assume that the commercial systems were all trained to detect hateful content or hate speech, while this model has not been trained on a single

[2] Google Jigsaw has since released a new version of the model powering the Perspective API (Lees et al., 2022). We assume that the new model would score higher on HateCheck.

[3] This procedure is equal to taking the argmax over *contradiction* and *entailment*.

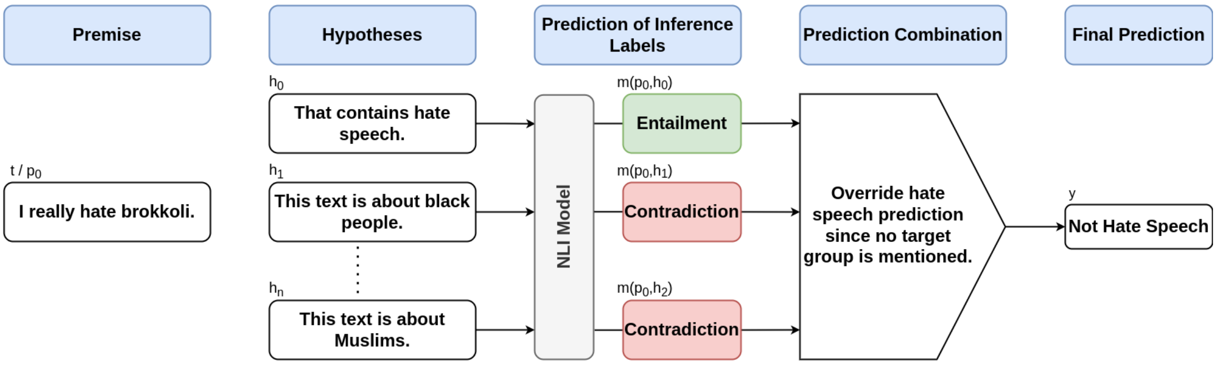
Premise
Hypotheses
Prediction of Inference Labels
Prediction Combination
Final Prediction
t / p0
I really hate brokkoli.
h0
That contains hate speech.
h1
This text is about black people.
hn
This text is about Muslims.
NLI Model
m(p0,h0)
Entailment
m(p0,h1)
Contradiction
m(p0,h2)
Contradiction
Override hate speech prediction since no target group is mentioned.
y
Not Hate Speech

Figure 1: **FBT** Standard zero-shot entailment predictions would wrongly predict the input text as containing hate speech. Using additional hypotheses it is possible to check if a protected group is targeted and if necessary to override the original prediction.

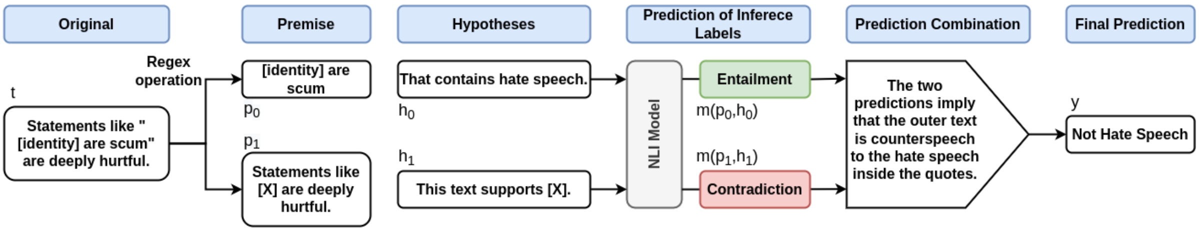
Original
Premise
Hypotheses
Prediction of Inferece Labels
Prediction Combination
Final Prediction
t
Statements like "[identity] are scum" are deeply hurtful.
Regex operation
[identity] are scum
p0
p1
Statements like [X] are deeply hurtful.
That contains hate speech.
h0
h1
This text supports [X].
NLI Model
Entailment
m(p0,h0)
m(p1,h1)
Contradiction
The two predictions imply that the outer text is counterspeech to the hate speech inside the quotes.
y
Not Hate Speech

Figure 2: **FCS** If a text contains quotations the quoted text is replaced with a variable X using a regular expression. Then, then two hypotheses are tested: The first hypothesis serves as a test checking if the text inside the quotes is hate speech. If that is predicted to be the case, the second hypothesis is used to predict if the quoted text is supported or denounced by the post.

example of hate speech detection or a similar task. The two lowest scoring hypotheses lead to an accuracy of 66.6% meaning that an unlucky choice of hypothesis can cost more than 12pp accuracy.

Error Analysis Column “No Strat.” in Table 4 shows the accuracy per HateCheck functionality for the hypothesis “That contains hate speech.”. Most notably, the model wrongly predicted all denouncements of hate (F20 and F21) as hate speech. In four functionalities (F22, F11, F23, F20) the model predicted hate speech even though no one or no relevant group was targeted. Finally, we see that the model often fails at analyzing sentences with negations (F15) and that it fails at recognizing when slurs are reclaimed and used in a positive way (F9). In what follows, we will present and evaluate strategies to avoid these errors.

5 Methods

In this section, we present four methods, which we call strategies, that aim to improve zero-shot hate speech detection. A strategy has the following components and structure: The aim is to assign a label $y = \{0, 1\}$ to input text t, where 1 corresponds to the class *hate speech* and 0 corresponds to the class *not-hate speech*. The input text t can be used in one or multiple a premises p_0 to p_m, that are used in conjunction with the main hypothesis h_0 and one or multiple supporting hypotheses $[h_1, ..., h_n]$ to obtain NLI model predictions $m(p_i, h_j) \in \{0, 1\}$ where 0 corresponds to contradiction and 1 corresponds to entailment. The variables i and j are defined as: $i \in [0, ..., m]$ and $j \in [0, ..., n]$. The rules for how to combine model predictions to obtain the final label y are given by the individual strategies. As the main hypothesis we use “That contains hate speech.”, since it lead to the highest accuracy on HateCheck in Section 4. The supporting hypotheses used to implement the strategies are listed in Table 3.

5.1 Filtering By Target (FBT)

The error analysis showed that we can improve zero-shot classification accuracy significantly by avoiding predictions of hate speech where no relevant target group occurs. We thus propose to avoid false positives by constructing a set of supporting hypotheses $[h_1, ..., h_n]$ to predict if text t actually targets or mentions a protected group or characteristic. If no protected group or characteristic is

FBT groups	This text is about women. This text is about trans people. This text is about gay people. This text is about black people. This text is about disabled people. This text is about Muslims. This text is about immigrants.
FBT characteristics	This text is about gender. This text is about sexual orientation. This text is about race. This text is about ethnicity. This text is about disability. This text is about religion. This text is about national origin.
FCS	This text supports [X].
FRS	This text is about myself.
CDC	This text is about insects. This text is about apes. This text is about primates. This text is about rats. This text is about a plague. This text has a negative sentiment.

Table 3: The supporting hypotheses used to implement the proposed strategies. For filtering by target we used the group-hypotheses for the HateCheck dataset and the characteristics-hypotheses for the ETHOS dataset, to account for differing hate speech definitions.

predicted to occur in t, a potential prediction of *hate speech* is overridden to *not-hate speech*. Figure 1 illustrates the method.

5.2 Filtering Counterspeech (FCS)

Our zero-shot model wrongly classifies all examples of counterspeech that quote or reference hate speech as actual hate speech. References to hate speech without quotation marks are hard to identify. Thus, for this work, we limit ourselves to counterspeech that quotes hate speech explicitly. We propose a three-stage strategy to this phenomenon: (1) quotation identification, (2) hate speech classification of the quoted content, (3) detecting the stance of the post towards the quoted content. Formally, the input text t is divided into premise p_0 which contains the quoted text and premise p_1 which contains the text around the quotes. The quoted text is represented as "$[X]$" in p_1. Using the main hypothesis h_0 we predict if p_0 contains hate speech or not. We use the supporting hypothesis "This text supports $[X]$." (h_1) to predict the stance of p_1 towards p_0. If p_0 contains hate speech and p_1 has a supportive stance towards p_0, t is classified as *hate speech*, otherwise it is classified as *not-hate speech*. The strategy is depicted in Figure 2.

5.3 Filtering Reclaimed Slurs (FRS)

As shown in Table 4, slurs that are reclaimed by members of a targeted group are often missclassified as hate speech. Based on the observation that a reclaimed slur is often ascribed to oneself, we propose to use a supporting hypothesis that indicates if text is self-directed.[4] If the model predicts self-directedness a potential prediction of *hate speech* is overridden to *not-hate speech*.

5.4 Catching Dehumanizing Comparisons (CDC)

One way of expressing hate towards a group and dehumanizing said group is to draw unflattering comparisons with animals. Such comparisons tend to be missed by hate speech detection systems, since the use of hateful or aggressive words is not needed to convey the hateful message. In HateCheck, this phenomenon is subsumed under "Implicit derogation". Standard zero-shot prediction obtains a moderately good accuracy of 89.3%. We test if false negatives can be caught with a three-step combination of supporting hypotheses: (1) use the supporting hypotheses of FBT to predict if a protected group is mentioned in text t, (2) predict if t has a negative sentiment, and (3) predict if t has is about animals typically used when making dehumanizing comparisons (such as insects, rats, or monkeys). If all conditions are met, override a prediction of *not-hate speech* to *hate speech*.

6 Experiments

We use the same model and adopt the entailment threshold of 0.5 from Section 4 for the main and all supporting hypotheses. Further, we take the hypothesis leading to the highest accuracy in Section 4 as the main hypothesis.

Since the main hypothesis in our experiments is chosen for maximum accuracy on HateCheck (based on the experiment in Section 4) and the strategies developed are based on an error analysis on HateCheck, the overall system might be overfitting on this specific dataset. An evaluation on this dataset might thus lead to results that overestimate a potential positive effect of the proposed strategies. We therefore also evaluate on ETHOS as an "unseen" dataset.

[4] Of course there are counterexamples to this rule, where reclaimed slurs are directed to others and not oneself. However, as long this approximation, as crude as it may be, helps to reduce false positives, it is a useful approximation.

Functionality	No Strat.	FBT	FCS	FRS	CDC	All
F1: Expression of strong negative emotions (explicit)	100.0	+0.0	+0.0	+0.0	+0.0	+0.0
F2: Description using very negative attributes (explicit)	98.6	+0.0	+0.0	+0.0	+0.0	+0.0
F3: Dehumanisation (explicit)	100.0	+0.0	+0.0	+0.0	+0.0	+0.0
F4: Implicit derogation	89.3	-5.0	+0.0	-10.0	+0.0	-12.9
F5: Direct threat	100.0	+0.0	+0.0	-3.0	+0.0	-3.0
F6: Threat as normative statement	99.3	+0.0	+0.0	+0.0	+0.0	+0.0
F7: Hate expressed using slur	85.4	-14.6	+0.0	+0.0	+2.8	-12.5
F8: Non-hateful homonyms of slurs	76.7	+6.7	+0.0	+0.0	+0.0	+6.7
F9: Reclaimed slurs	33.3	+0.0	+0.0	+32.1	+0.0	+32.1
F10: Hate expressed using profanity	97.9	-0.7	+0.0	+0.0	+0.0	-0.7
F11: Non-hateful use of profanity	43.0	+49.0	+0.0	+23.0	+0.0	+50.0
F12: Hate expressed through reference in subsequent clauses	100.0	+0.0	+0.0	-2.9	+0.0	-2.9
F13: Hate expressed through reference in subsequent sentences	97.7	+0.0	+0.0	+0.0	+0.0	+0.0
F14: Hate expressed using negated positive statement	100.0	-2.9	+0.0	+0.0	+0.0	-2.9
F15: Non-hate expressed using negated hateful statement	33.1	+5.3	+0.0	+0.0	+0.0	+5.3
F16: Hate phrased as a question	99.3	+0.0	+0.0	-5.0	+0.0	-5.0
F17: Hate phrased as an opinion	100.0	+0.0	+0.0	-2.3	+0.0	-2.3
F18: Neutral statements using protected group identifiers	96.0	+0.0	+0.0	+0.0	+0.0	+0.0
F19: Positive statements using protected group identifiers	97.4	+0.0	+0.0	+0.0	+0.0	+0.0
F20: Denouncements of hate that quote it	0.0	+8.7	+100.0	+0.0	+0.0	+100.0
F21: Denouncements of hate that make direct reference to it	0.0	+7.8	+0.0	+1.4	+0.0	+8.5
F22: Abuse targeted at objects	63.1	+36.9	+0.0	+9.2	+0.0	+36.9
F23: Abuse targeted at individuals (not as member of a prot. group)	7.7	+70.8	+0.0	+0.0	+0.0	+70.8
F24: Abuse targeted at nonprotected groups (e.g. professions)	11.3	+83.9	+0.0	+3.2	+0.0	+83.9
F25: Swaps of adjacent characters	97.7	+0.0	+0.0	+0.0	+0.0	+0.0
F26: Missing characters	88.6	-1.4	+0.0	+0.0	+0.7	-0.7
F27: Missing word boundaries	87.9	-4.3	+0.0	+0.0	+1.4	-3.5
F28: Added spaces between chars	97.7	-11.0	+0.0	-0.6	+0.0	-11.6
F29: Leet speak spellings	93.1	-12.7	+0.0	+0.0	+0.6	-12.1
Overall	79.4	+3.3	+4.6	+0.7	+0.2	+7.9

Table 4: Analysis of how individual functionalities are affected by the proposed strategies. The functionality descriptions are taken from Röttger et al. (2021). *No Strat.* refers to using only the hypothesis “That contains hate speech.”. Accuracies below 70% are marked in red. *All* refers to combining all four proposed strategies. The columns *FBT*, *FCS*, *FRS*, *CDC* and *All* contain the difference in percentage point (pp) accuracy compared to *No Strat.*.

ETHOS does not refer to protected groups in its definition and annotation of hate speech, but instead to protected characteristics. Thus, in the hypotheses for FBT we replace protected groups with the protected characteristics listed in Table 3.

6.1 Results

HateCheck The bottom row *Overall* in Table 4 shows the results for the proposed strategies and their combination on the HateCheck dataset. Each strategy leads to an improvement in accuracy. But while FBT and FCS lead to large increases, FRS and CDC only lead to minor increases. Combining all proposed strategies leads to an increase in accuracy of 7.9pp.

ETHOS The results of evaluating the same strategies on ETHOS (Mollas et al., 2022) are given in Table 5. As additional baselines compared to zero-shot prediction using just one hypothesis, we include the performance of three models trained on ETHOS by Mollas et al. (2022).

strategies	accuracy (%)	Δ
(ETHOS) SVM	66.4	-
(ETHOS) BERT	80.0	-
(ETHOS) DistilBERT	80.4	-
“That contains hate speech.”	69.6	+0.0
FBT (TG)	75.5	+5.9
FBT (TC)	78.7	+9.1
FCS	69.6	+0.0
FRS	71.3	+1.7
CDC (TC)	69.5	-0.1
FBT (TC) + FCS	78.7	+9.1
FBT (TC) + FRS	79.7	+10.1
FCS + FRS	71.3	+1.7
FBT (TC) + FCS + FRS	79.7	+10.1
CDC (TC) + FBT (TC) + FCS + FRS	79.6	+10.0

Table 5: Accuracy scores on ETHOS. The three top rows show baselines computed by Mollas et al. (2022). TG refers to using *target groups* to implement FBT and TC refers to using *target characteristics* for FBT.

The combination of all strategies leads to a increase of 10.0pp, which is an even greater increase than on HateCheck. However, the gains are more unevenly distributed across the proposed strategies. Filtering by target characteristics alone leads to an increase of 9.1pp. Filtering reclaimed slurs still has a positive effects of 1.7pp. However, filtering counterspeech, the best performing strategy on HateCheck, does not have any effect at all. And

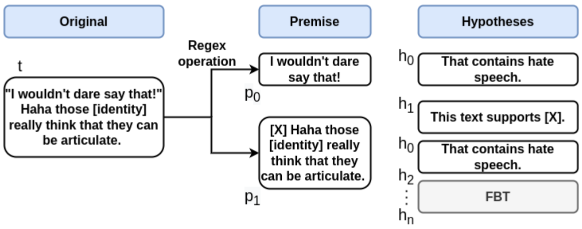
Original
Premise
Hypotheses
t
"I wouldn't dare say that!" Haha those [identity] really think that they can be articulate.
Regex operation
I wouldn't dare say that!
p_0
[X] Haha those [identity] really think that they can be articulate.
p_1
h_0
That contains hate speech.
h_1
This text supports [X].
h_0
That contains hate speech.
h_2
FBT
h_n

Figure 3: Counterspeech filter adjusted for detecting hate speech where quotes are present but the hate speech is outside of the quote - i.e. in the outer text.

catching dehumanizing comparisons even reduces performance by 0.1pp.

The comparison to the baselines provided by Mollas et al. (2022) shows that zero-shot prediction using the hypothesis "That contains hate speech." already outperforms a trained SVM by more than 3pp but still underperforms the fine-tuned BERT by more than 10pp. However, applying the proposed strategies almost closes the gap to the fine-tuned models.

6.2 Analysis of Affected Functionalities

We analyse if the observed performance gains actually stem from improvements on the functionalities targeted by the proposed strategies. Table 4 shows for each functionality how it was affected by each strategy.

Filtering by Target The results for filtering by target show dramatic accuracy increases for HateCheck functionalities containing abuse and profanity that is not targeted at a protected group. These are exactly the functionalities this strategy aimed at. The performance for spelling variations and implicit derogation decreases slightly. This can be explained by the model failing to correctly recognize spelling variations of target groups and by the fact that the target group might only be implied in implicit derogation leading to false negatives.

Filtering Counterspeech The counterspeech filter increases the accuracy of the respective functionality from 0% to 100%. Thus, detecting quoted hate speech as well as detecting the stance towards the quote worked exactly as intented on HateCheck.

Filtering Reclaimed Slurs The functionality with the largest gains when filtering reclaimed slurs is "reclaimed slurs", showing that the strategy works as intended. However, the performance increase of this method is not as high as for example filtering by target. The functionality "non-hateful use of profanity" also benefits from this strategy. We assume that such uses of profanity often are also not directed at other people and thus sometimes predicted to be directed at oneself. This is a beneficial side-effect of the strategy.

Catching Dehumanizing Comparisons This strategy only leads to minor a minor overall improvement of 0.2pp. We observe no effect on the targeted functionality, but a small positive effect on F7, "Hate expressed using slur", which could indicate that the model associates slurs with negatively coded animals. Additionally, the strategy has minor positive effects on functionalities that contain spelling variations.

7 Discussion

Supporting Hypotheses The performance of NLI-strategies largely depends on the accuracy of the supporting hypotheses. Testing the accuracy of each supporting hypothesis is not always possible, since annotated data for the predicted aspect of the input text might not be available. Indeed, one of the strengths of our approach is that it can use aspects for which no annotated data exists. Another uncertainty lies in the formulation of supporting hypotheses. A suboptimal formulation of supporting hypotheses negatively affects the overall results. By using annotated targets in HateCheck and inferring stance labels as well as self-directedness from HateCheck functionalities, we compute and compare the accuracy of multiple supporting hypothesis formulations. The results (in Appendix B) show that testing for the presence of a target group mostly leads to accuracies above 90%, independent of the specific formulation. Detecting the outer stance towards the inner text obtained a perfect accuracy of 100% and testing for self-directedness leads to low accuracies, which are probably partly due to faulty label inferences from functionalities. Overall, the results indicate that the supporting hypotheses provide reliable information.

Generality There are two ways in which the proposed strategies might not generalize. First, the strategies might be specific to the model used for the experiments. In order to answer this question, repeating the experiments with other NLI models will be necessary. Second, the strategies might be specific to HateCheck and not generalize to other datasets, since they specifically target HateCheck functionalities. The evaluation on ETHOS shows

strategy	F20	overall
No strategy	0.0	79.4
FCS	+100	+4.6
FCS$_{p_1}$	+0.0	+0.0
FCS$_{p_1 FBT}$	+6.9	+0.3

Table 6: Evaluation of FCS variants. The two bottom rows display the variants adjusted for detecting Hate Speech in p_1. The functionality F20 contains *Denouncements of hate that quote it*. The scores are given in accuracy (%) and change in accuracy compared to *No strategy*.

that this is generally not case, since the results for the best strategy combination on ETHOS even exceed the results for the best combination on HateCheck.

While our experiments did not show problems in generalization, we can imagine the following weakness for the FCS strategy: Given an input text t that contains a quote and hate speech, where the hate speech does not occur inside of the quotes, the current FCS strategy would fail, since it only detects hate speech in p_0, that is inside the quotes. Such an example is given in Figure 3.

The obvious solution to avoid this problem is to not only apply the main hypothesis h_0 on the premise p_0 but also on premise p_1, which contains the text outside the quotes. In a follow-up experiment we implement this modified strategy (FCS$_{p_1}$) and evaluated it on HateCheck. Note, that since such a case is not covered by HateCheck or ETHOS there is no increase in accuracy to be expected - we can only test if accounting for this case leads to a decrease in accuracy through unwanted side effects.

The results, displayed in Table 6 show that this modification removes all the gains obtained through FBT. We assume that this is due to the fact that the counterspeech often also conveys strong negative emotions that are mistaken by the model for hate speech.

We further test if this problem can be alleviated by applying the FBT strategy if hate speech is detected in p_1 (i.e. outside of the quotes) as depicted in Figure 3. The results in Table 6 (row FCS$_{p_1 FBT}$) show that additionally applying FBT only recovers a fraction of the positive effect of FCS. We assume that this is due to counterspeech including or being associated with target groups. Thus, further research that investigates how the problem can be alleviated is needed.

Efficiency In the proposed setup each new hypothesis necessitates an additional forward pass, which means that the computational cost linearly increases with adding new hypotheses. This leads to a difficult trade-off between accuracy and efficiency. A possible solution was recently proposed by Müller et al. (2022), who embed premises and hypotheses independently, thereby keeping the computational cost during inference time with respect to the number of hypotheses constant.

Prerequisites of FBT FBT presumes that the target groups or characteristics are known beforehand. This prerequisite is unproblematic when using FBT to detect hate speech against well known targets of hate speech or discrimination. However, it makes this method unsuitable for tasks such as vulnerable group identification (Mossie and Wang, 2020).

Flexibility Single hypotheses or entire strategies can be easily added to or removed from a system. This modularity makes the approach easily adjustable to different scenarios or use cases. For example, if precision is the main concern, the *catching dehumanizing comparisons* can be dropped and if recall is the main concern, filters can be removed. Instead of adding or removing strategies, it is also possible to manipulate the precision-recall trade-off by adjusting confidence thresholds for particular hypotheses.

8 Conclusion

In this work, we combine hypotheses to create more accurate NLI-based zero-shot hate speech detection systems. Specifically, we develop four simple strategies, *filtering by target*, *filtering counter speech*, *filtering reclaimed-slurs*, and *catching dehumanizing comparisons*, that target specific model weaknesses. We evaluate the strategies on HateCheck, which served as the basis for developing these strategies, and on ETHOS, which acts as an "unseen" dataset. The NLI-based zero-shot baseline already outperforms fine-tuned models on HateCheck and beats an SVM baseline on ETHOS. Using all four proposed strategies leads to a further performance increase of 7.9% on HateCheck and 10.0% on ETHOS. However, the contribution of the strategies to the performance increases varies, with *catching dehumanizing comparisons* even having a small negative effect on the accuracy on ETHOS.

The proposed approach is simple and modular making it easy to implement and adjust to different scenarios.

In future work, we plan to evaluate such strategies in a multi-lingual setup and in a few-shot scenario. Further, this works leads to the question how effective supporting hypotheses could be searched and generated automatically.

Acknowledgements

We thank Chantal Amrhein, Jonathan Schaber, Aneta Zuber, and the anonymous reviewers for their helpful feedback. This work was funded by the University of Zurich Research Priority Program (project "URPP Digital Religion(s)"[5]).

Ethical Considerations

The goal of this article is to contribute to the development of sophisticated hate speech detection methods, and thus to contribute to an online environment that is less hateful. However, we can imagine multiple ways how such methods, and our proposed approach in particular, can lead to harm: (1) Deploying the exact system that we propose would lead to not detecting hate speech against protected groups that are not explicitly included in the two datasets we worked with and thus not covered by the FBT-method. Thus, before deploying such a method, careful consideration of which protected groups or group characteristics are covered is needed. (2) Overconfident claims about the accuracy of hate speech detection methods could lead to the false impression that content moderation can be left to automatic methods with no human intervention. (3) Hate speech detection in general is prone to misuse and repurposing in order to prohibit other kinds of speech. Detecting if a text revolves around a protected group could be misused to detect and prohibit important discussion around topics connected to a protected group.

However, we believe that a decomposition of hate speech into more specific aspects is important for more accurate, interpretable and modular hate speech detection methods (Khurana et al., 2022). Thus, detecting such components of hate speech is also strongly beneficial for effective content moderation and a less hateful online environment.

[5] https://www.digitalreligions.uzh.ch/en.html

References

Badr AlKhamissi, Faisal Ladhak, Srini Iyer, Ves Stoyanov, Zornitsa Kozareva, Xian Li, Pascale Fung, Lambert Mathias, Asli Celikyilmaz, and Mona Diab. 2022. ToKen: Task Decomposition and Knowledge Infusion for Few-Shot Hate Speech Detection. arXiv. ArXiv ID: arXiv:2205.12495 [cs].

Somnath Banerjee, Maulindu Sarkar, Nancy Agrawal, Punyajoy Saha, and Mithun Das. 2021. Exploring Transformer Based Models to Identify Hate Speech and Offensive Content in English and Indo-Aryan Languages. *arXiv:2111.13974 [cs]*. ArXiv: 2111.13974.

Ke-Li Chiu and Rohan Alexander. 2021. Detecting Hate Speech with GPT-3. *arXiv:2103.12407 [cs]*. ArXiv: 2103.12407.

Sourav Das, Prasanta Mandal, and Sanjay Chatterji. 2021. Probabilistic Impact Score Generation using Ktrain-BERT to Identify Hate Words from Twitter Discussions. *arXiv:2111.12939 [cs]*. ArXiv: 2111.12939.

Thomas Davidson, Dana Warmsley, 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. Number: 1.

Mai ElSherief, Caleb Ziems, David Muchlinski, Vaishnavi Anupindi, Jordyn Seybolt, Munmun De Choudhury, and Diyi Yang. 2021. Latent Hatred: A Benchmark for Understanding Implicit Hate Speech. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 345–363, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Komal Florio, Valerio Basile, Marco Polignano, Pierpaolo Basile, and Viviana Patti. 2020. Time of Your Hate: The Challenge of Time in Hate Speech Detection on Social Media. *Applied Sciences*, 10(12):4180. Number: 12 Publisher: Multidisciplinary Digital Publishing Institute.

Paula Fortuna and Sérgio Nunes. 2018. A Survey on Automatic Detection of Hate Speech in Text. *ACM Computing Surveys*, 51(4):85:1–85:30.

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*.

Md Saroar Jahan and Mourad Oussalah. 2021. A systematic review of Hate Speech automatic detection using Natural Language Processing. Technical Report arXiv:2106.00742, arXiv. ArXiv:2106.00742 [cs] type: article.

Prashant Kapil and Asif Ekbal. 2020. A deep neural network based multi-task learning approach to hate speech detection. *Knowledge-Based Systems*, 210:106458.

Urja Khurana, Ivar Vermeulen, Eric Nalisnick, Marloes Van Noorloos, and Antske Fokkens. 2022. Hate Speech Criteria: A Modular Approach to Task-Specific Hate Speech Definitions. In *Proceedings of the Sixth Workshop on Online Abuse and Harms (WOAH)*, pages 176–191, Seattle, Washington (Hybrid). Association for Computational Linguistics.

Eric Lavergne, Rajkumar Saini, György Kovács, and Killian Murphy. 2020. TheNorth @ HaSpeeDe 2: BERT-based Language Model Fine-tuning for Italian Hate Speech Detection. In Valerio Basile, Danilo Croce, Maria Maro, and Lucia C. Passaro, editors, *EVALITA Evaluation of NLP and Speech Tools for Italian - December 17th, 2020*, pages 142–147. Accademia University Press.

Alyssa Lees, Vinh Q. Tran, Yi Tay, Jeffrey Sorensen, Jai Gupta, Donald Metzler, and Lucy Vasserman. 2022. A New Generation of Perspective API: Efficient Multilingual Character-level Transformers. *arXiv:2202.11176 [cs]*. ArXiv: 2202.11176.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation, and Comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880, Online. Association for Computational Linguistics.

Pengfei Liu, Weizhe Yuan, Jinlan Fu, Zhengbao Jiang, Hiroaki Hayashi, and Graham Neubig. 2021. Pre-train, Prompt, and Predict: A Systematic Survey of Prompting Methods in Natural Language Processing. *arXiv:2107.13586 [cs]*. ArXiv: 2107.13586.

Thomas Mandl, Sandip Modha, Anand Kumar M, and Bharathi Raja Chakravarthi. 2020. Overview of the HASOC Track at FIRE 2020: Hate Speech and Offensive Language Identification in Tamil, Malayalam, Hindi, English and German. In *Forum for Information Retrieval Evaluation*, pages 29–32, Hyderabad India. ACM.

Thomas Mandl, Sandip Modha, Prasenjit Majumder, Daksh Patel, Mohana Dave, Chintak Mandlia, and Aditya Patel. 2019. Overview of the HASOC Track at FIRE 2019: Hate Speech and Offensive Content Identification in Indo-European Languages. In *Proceedings of the 11th Forum for Information Retrieval Evaluation*, FIRE '19, pages 14–17, New York, NY, USA. Association for Computing Machinery. Event-place: Kolkata, India.

Ioannis Mollas, Zoe Chrysopoulou, Stamatis Karlos, and Grigorios Tsoumakas. 2022. ETHOS: a multi-label hate speech detection dataset. *Complex & Intelligent Systems*.

Zewdie Mossie and Jenq-Haur Wang. 2020. Vulnerable community identification using hate speech detection on social media. *Information Processing & Management*, 57(3):102087.

Thomas Müller, Guillermo Pérez-Torró, and Marc Franco-Salvador. 2022. Few-Shot Learning with Siamese Networks and Label Tuning. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8532–8545, Dublin, Ireland. Association for Computational Linguistics.

Preslav Nakov, Vibha Nayak, Kyle Dent, Ameya Bhatawdekar, Sheikh Muhammad Sarwar, Momchil Hardalov, Yoan Dinkov, Dimitrina Zlatkova, Guillaume Bouchard, and Isabelle Augenstein. 2021. Detecting Abusive Language on Online Platforms: A Critical Analysis. *arXiv:2103.00153 [cs]*. ArXiv: 2103.00153.

Huy Nghiem and Fred Morstatter. 2021. "Stop Asian Hate!" : Refining Detection of Anti-Asian Hate Speech During the COVID-19 Pandemic. *arXiv:2112.02265 [cs]*. ArXiv: 2112.02265.

Debora Nozza. 2021. Exposing the limits of Zero-shot Cross-lingual Hate Speech 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 2: Short Papers)*, pages 907–914, Online. Association for Computational Linguistics.

Endang Wahyu Pamungkas, Valerio Basile, and Viviana Patti. 2021. A joint learning approach with knowledge injection for zero-shot cross-lingual hate speech detection. *Information Processing & Management*, 58(4):102544.

Flor Miriam Plaza-Del-Arco, M. Dolores Molina-González, L. Alfonso Ureña-López, and María Teresa Martín-Valdivia. 2021. A Multi-Task Learning Approach to Hate Speech Detection Leveraging Sentiment Analysis. *IEEE Access*, 9:112478–112489. Conference Name: IEEE Access.

Fabio Poletto, Valerio Basile, Manuela Sanguinetti, Cristina Bosco, and Viviana Patti. 2021. Resources and benchmark corpora for hate speech detection: a systematic review. *Language Resources and Evaluation*, 55(2):477–523.

Rahul Pradhan, Ankur Chaturvedi, Aprna Tripathi, and Dilip Kumar Sharma. 2020. A Review on Offensive Language Detection. In *Advances in Data and Information Sciences*, Lecture Notes in Networks and Systems, pages 433–439, Singapore. Springer.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer. *Journal of Machine Learning Research*, 21(140):1–67.

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.

Niloofar Safi Samghabadi, Parth Patwa, Srinivas PYKL, Prerana Mukherjee, Amitava Das, and Thamar Solorio. 2020. Aggression and Misogyny Detection using BERT: A Multi-Task Approach. In *Proceedings of the Second Workshop on Trolling, Aggression and Cyberbullying*, pages 126–131, Marseille, France. European Language Resources Association (ELRA).

Punyajoy Saha, Binny Mathew, Pawan Goyal, and Animesh Mukherjee. 2018. Hateminers : Detecting Hate speech against Women. *arXiv:1812.06700 [cs]*. ArXiv: 1812.06700.

Maarten Sap, Saadia Gabriel, Lianhui Qin, Dan Jurafsky, Noah A. Smith, and Yejin Choi. 2020. Social Bias Frames: Reasoning about Social and Power Implications of Language. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5477–5490, Online. Association for Computational Linguistics.

Timo Schick, Sahana Udupa, and Hinrich Schütze. 2021. Self-Diagnosis and Self-Debiasing: A Proposal for Reducing Corpus-Based Bias in NLP. *Transactions of the Association for Computational Linguistics*, 9:1408–1424.

Lukas Stappen, Fabian Brunn, and Björn Schuller. 2020. Cross-lingual Zero- and Few-shot Hate Speech Detection Utilising Frozen Transformer Language Models and AXEL. *arXiv:2004.13850 [cs, stat]*. ArXiv: 2004.13850.

Moshe Uzan and Yaakov HaCohen-Kerner. 2021. Detecting Hate Speech Spreaders on Twitter using LSTM and BERT in English and Spanish. *CLEF 2021 – Conference and Labs of the Evaluation Forum, September 21–24, 2021, Bucharest, Romania*, CEUR Workshop Proceedings:8.

Sinong Wang, Han Fang, Madian Khabsa, Hanzi Mao, and Hao Ma. 2021. Entailment as Few-Shot Learner. *arXiv:2104.14690 [cs]*. ArXiv: 2104.14690.

Zeerak Waseem. 2016. Are You a Racist or Am I Seeing Things? Annotator Influence on Hate Speech Detection on Twitter. In *NLP+CSS@EMNLP*.

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, San Diego, California. Association for Computational Linguistics.

Adina Williams, Nikita Nangia, and Samuel Bowman. 2018. A Broad-Coverage Challenge Corpus for Sentence Understanding through Inference. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1112–1122, New Orleans, Louisiana. Association for Computational Linguistics.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. 2020. Transformers: State-of-the-Art Natural Language Processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

Wenjie Yin and Arkaitz Zubiaga. 2021. Towards generalisable hate speech detection: a review on obstacles and solutions. *PeerJ Computer Science*, 7:e598. Publisher: PeerJ Inc.

Wenpeng Yin, Jamaal Hay, and Dan Roth. 2019. Benchmarking Zero-shot Text Classification: Datasets, Evaluation and Entailment Approach. 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 3914–3923, Hong Kong, China. Association for Computational Linguistics.

A Zero-Shot Results: Comparing Hypotheses

Table 7, the extended version of Table 2, contains all results for comparing hypotheses for zero-shot hate speech detection on HateCheck. “average:<`expression`>” refers to the average accuracy of all hypotheses containing `expression`. The highest accuracy is in bold.

hypothesis	accuracy (%)
Containing hate speech.	74.7
Contains hate speech.	78.6
Hate speech.	72.9
Hateful.	71.8
It contains hate speech.	78.7
It is hateful.	75.0
It contains hate speech.	78.7
It is hate speech.	70.8
It is hateful.	75.0
That contains hate speech.	**79.4**
That contains hateful content.	78.0
That example contains hateful content.	77.8
That example is hate speech.	66.6
That example is hateful.	76.8
That is hateful.	66.6
That text contains hate speech.	78.8
That text contains hateful content.	78.6
That text is hate speech.	69.2
That text is hateful.	77.2
This contains hate speech.	79.1
This contains hateful content.	78.2
This example contains hate speech.	77.3
This example contains hateful content.	77.8
This example is hate speech.	67.2
This example is hateful.	77.4
This is hateful.	70.6
This text contains hate speech.	78.8
This text contains hateful content.	78.3
This text is hate speech.	69.5
This text is hateful.	78.7
average: It	74.8
average: This	75.7
average: That	74.9
average: hateful	75.8
average: hateful content	78.1
average: hate speech	74.5
average: example	74.4
average: text	76.1
average: is	73.9
average: contain	78.2
SiftNinja	33.2
BERT fine-tuned on Davidson et al. (2017)	60.2
BERT fine-tuned on Founta et al. (2018)	63.2
Google-Jigsaw	76.6

Table 7: Full evaluation of hypotheses, that claim hate speech exists in the input text, on HateCheck.

B Evaluating Supporting Hypotheses

B.1 Target Groups and Target Characteristics

Each example in HateCheck which mentions a protected group or revolves around a protected group is annotated with said group. If no group is targeted the example is annotated with an empty string. By using these annotations as labels, we can create a binary classification task for each protected group: for detecting a mention of a specific protected group x, we convert label x to 1 and all other labels (i.e. all other protected groups) to 0.

We use the same model and as in the previous zero-shot experiments for evaluation and test the performance for detecting mentions for all protected groups in HateCheck. We additionally test the detection of the supercategory *queer people* covering the two protected groups *gay people* and *transgender people* in HateCheck. When testing if a text revolves around gender, we treat both *women* and *transgender people* as positive classes and all other protected groups as negative classes. While this mapping obviously can result in incorrect labels (a text can be about gender even if another group is targeted), we assume that it holds true for examples in the HateCheck dataset.

Table 8 shows the results for detecting if *black people* are mentioned, Table 9 for mentions of *Muslims*, Table 10 for mentions of immigrants, Table 11 for mentions of *disabled people*, Table 12 for mentions of *gay people*, Table 13 for mentions of *transgender people*, Table 14 for mentions of *queer people*, and Table 16 for detecting if a text is about *gender*.

The results show that in many cases the detection of a mentioned group is surprisingly accurate. The difference in accuracy between the best performing hypothesis and the worst performing hypothesis does not exceed 12%. This is a similar range to the differences found between hypotheses when testing if a text contains hate speech (see Table 2 and Table 7). However, when looking at F_1 scores the differences are much larger, with more general terms, such as *faith* or *ethnicity* preforming worse than the specific terms *Muslims* and *black people*.

Detecting if a text revolves around gender performs worst, compared to detecting other protected groups or characteristics. This is mostly due to low precision scores. We assume that this is a consequence of sexual orientation (*gay people*) being closely associated in the embedding space with *gender* and thus leading to false positives.

B.2 Self-Directedness

Evaluating the accuracy of detecting self-directedness is difficult, because there exist no

hypothesis	accuracy (%)	$\downarrow F_1$ (%)	recall (%)	precision (%)
That example is about black people.	97.9	92.0	93.8	90.2
This example is about black people.	97.6	90.9	94.0	88.0
That text is about black people.	96.4	87.1	93.2	81.8
That is about black people.	95.8	85.0	92.5	78.7
This text is about black people.	95.2	83.6	94.4	75.0
This is about black people.	95.0	83.0	93.6	74.5
That example is about people of colour.	94.3	80.7	92.7	71.4
That example is about race.	94.0	79.8	91.3	70.9
This example is about people of colour.	93.8	79.3	92.3	69.5
That is about race.	94.6	77.5	72.6	83.1
This example is about race.	90.9	72.3	91.7	59.6
That text is about people of colour.	89.8	70.6	94.2	56.4
That example is about ethnicity.	90.5	69.8	85.3	59.1
This is about race.	88.9	67.6	89.4	54.4
That text is about race.	87.8	66.3	92.3	51.7
That is about people of colour.	87.4	65.9	93.8	50.8
This text is about race.	86.5	64.5	94.6	48.9
This text is about people of colour.	84.9	62.1	96.1	45.9
This example is about ethnicity.	85.6	62.1	91.1	47.1
This is about ethnicity.	85.4	58.1	78.4	46.2
That text is about ethnicity.	81.5	56.1	91.3	40.5
This text is about ethnicity.	80.2	55.0	93.6	38.9
This is about people of colour.	74.8	49.4	95.2	33.3
That is about ethnicity.	87.0	28.4	19.9	49.7

Table 8: Results for supporting hypotheses aimed at detecting mentions of black people. The hypotheses are sorted by macro F_1-score in descending order. Note, that some of the hypotheses listed use broader terms ("people of colour", "race", "ethnicity") that should also detect the mentions of other target groups. However, in the context of HateCheck, we can only test the detection of mentioning black people.

labels in HateCheck that could be used as a ground truth.

One possibility, that follows the motivation for introducing the FRS-method, is to treat all examples of functionality F9 ("reclaimed_slur") as self-directed and examples of all other functionalities as not self-directed. We conducted this experiment. The results are given in Table 17. However, one should keep in mind that this disregards that reclaimed slurs can be used in a not-self directed manner and that other functionalities, such as functionality F11 *non-hateful use of profanity*, might contain examples of self-directed speech.

B.3 Counterspeech

We perform a simple evaluation of the supporting hypothesis that predicts the stance of an outer text towards its quoted inner text (see Section 5.2 for an explanation) using only functionality F20 (*Denouncements of hate that quote it*) as an evaluation set. We treat stance detection here as a binary task with the labels *is_for* or *is_against*.

How these labels are mapped onto NLI labels depends on the specific hypothesis. If the hypothesis claims that the outer text supports the quoted text, then *is_for* is mapped to *entailment* and *is_against* is mapped to *contradiction*. Conversely, if the hypothesis claims that the outer text denounces the quoted text, then *is_for* is mapped to *contradiction* and *is_against* is mapped to *entailment*.

We test various formulations including the verbs "supports [X]" and "is for [X]". The results are given in Table 18. Since all examples in this category are considered hate speech, that is denounced by the outer text, the true label is always *is_against*. We only report accuracy, since there can be no false positives and true negatives, which makes precision and recall lose its usefulness.

hypothesis	accuracy (%)	↓ F_1 (%)	recall (%)	precision (%)
That example is about Muslims.	98.3	93.1	90.7	95.6
This example is about Muslims.	98.1	92.7	90.7	94.8
This text is about Muslims.	98.1	92.6	90.3	95.0
That text is about Muslims.	98.0	92.3	90.3	94.4
That example is about Muslim people.	98.0	92.0	90.7	93.4
This example is about Muslim people.	97.9	92.0	90.9	93.0
This is about Muslims.	97.8	91.6	90.7	92.4
This text is about Muslim people.	97.8	91.5	90.7	92.2
That text is about Muslim people.	97.7	91.1	90.1	92.2
This example is about religion.	97.7	90.3	83.3	98.5
This text is about religion.	97.5	89.7	83.9	96.4
This is about Muslim people.	97.3	89.6	89.9	89.3
That text is about religion.	97.5	89.5	82.9	97.3
That is about Muslims.	97.2	89.1	88.6	89.6
That example is about religion.	97.3	88.5	80.2	98.7
That is about Muslim people.	96.8	87.8	89.7	85.9
This is about religion.	96.4	84.5	74.8	97.1
This example is about faith.	95.0	78.2	69.2	89.8
This text is about faith.	95.0	78.1	69.0	90.0
That text is about faith.	94.4	74.2	62.4	91.5
That example is about faith.	93.4	68.9	56.6	88.1
This is about faith.	92.0	63.3	52.9	78.8
That is about religion.	89.6	34.5	21.1	95.3
That is about faith.	88.3	21.9	12.6	82.4

Table 9: Results for supporting hypotheses aimed at detecting mentions of Muslim people. Note, that some of the hypotheses listed use broader terms ("faith", "religion") that should detect other target groups too. However, in the context of HateCheck their applicability is restricted to Muslim people, since no other religion occurs in HateCheck.

hypothesis	accuracy (%)	↓ F_1 (%)	recall (%)	precision (%)
That example is about immigrants.	97.8	91.2	92.4	89.9
This example is about immigrants.	97.7	90.8	92.4	89.2
That is about immigrants.	97.2	88.8	89.2	88.4
That text is about immigrants.	97.0	88.6	92.2	85.2
This text is about immigrants.	96.3	86.4	93.7	80.1
This is about immigrants.	96.4	86.2	91.6	81.4
This text is about national origin.	77.7	42.2	65.4	31.1
That text is about national origin.	78.0	41.3	62.4	30.9
This is about national origin.	83.4	37.0	39.3	35.0
That example is about national origin.	81.4	30.0	32.2	28.1
This example is about national origin.	79.8	30.0	34.8	26.3
That is about national origin.	86.5	24.6	17.7	40.2

Table 10: Results for supporting hypotheses aimed at detecting mentions of immigrants.

hypothesis	accuracy (%)	↓ F_1 (%)	recall (%)	precision (%)
That example is about disabled people.	98.4	93.7	90.3	97.3
This example is about disabled people.	98.4	93.5	90.1	97.1
This text is about disabled people.	98.0	92.3	91.3	93.2
That example is about disability.	97.9	91.9	92.4	91.4
That text is about disabled people.	97.9	91.3	87.0	96.1
This example is about disability.	97.7	91.3	93.0	89.6
That is about disabled people.	97.7	90.9	87.2	94.8
This is about disabled people.	97.4	89.9	90.3	89.5
That text is about disability.	96.6	87.6	91.5	84.1
That is about disability.	96.6	86.0	80.4	92.4
This is about disability.	94.8	82.0	91.7	74.1
This text is about disability.	94.5	81.6	94.4	71.9

Table 11: Results for supporting hypotheses aimed at detecting mentions of disabled people.

hypothesis	accuracy (%)	↓ F_1 (%)	recall (%)	precision (%)
That example is about gay people.	99.1	96.7	94.2	99.4
This example is about gay people.	99.0	96.6	94.0	99.2
This text is about gay people.	98.8	95.9	92.9	99.0
This is about gay people.	98.8	95.8	92.6	99.2
That text is about gay people.	98.5	94.6	90.0	99.6
That is about gay people.	98.4	94.4	90.4	98.8

Table 12: Results for supporting hypotheses aimed at detecting mentions of gay people.

hypothesis	accuracy (%)	↓ F_1 (%)	recall (%)	precision (%)
That example is about transgender people.	99.0	95.8	92.7	99.1
That text is about transgender people.	99.0	95.6	92.2	99.3
This text is about transgender people.	98.9	95.6	92.9	98.4
This example is about transgender people.	98.9	95.4	92.2	98.8
That is about transgender people.	98.8	94.9	92.0	97.9
This is about transgender people.	98.7	94.6	92.2	97.0

Table 13: Results for supporting hypotheses aimed at detecting mentions of transgender people.

hypothesis	accuracy (%)	↓ F_1 (%)	recall (%)	precision (%)
This is about queer people.	94.3	88.6	81.3	97.5
That example is about queer people.	93.7	87.1	77.9	98.8
This example is about queer people.	93.2	85.9	76.2	98.5
This text is about queer people.	93.1	85.7	76.4	97.5
That is about queer people.	92.4	84.0	73.5	98.2
That text is about queer people.	91.7	82.3	70.8	98.4

Table 14: Results for supporting hypotheses aimed at detecting mentions of queer people, which in HateCheck corresponds to the categories *gay people* and *transgender people*.

hypothesis	accuracy (%)	↓ F_1 (%)	recall (%)	precision (%)
This example is about women.	97.2	90.2	94.1	86.6
That example is about women.	97.2	90.1	94.1	86.5
That is about women.	96.2	86.6	88.6	84.6
This text is about women.	95.9	86.1	93.9	79.5
That text is about women.	95.7	85.3	91.6	79.8
This is about women.	94.8	83.0	92.7	75.0

Table 15: Results for supporting hypotheses aimed at detecting mentions of women.

hypothesis	accuracy (%)	↓ F_1 (%)	recall (%)	precision (%)
That example is about gender.	90.0	81.8	85.8	78.2
This example is about gender.	89.0	80.5	87.0	74.9
That text is about gender.	88.4	79.6	86.8	73.5
This text is about gender.	87.9	79.3	88.9	71.5
This is about gender.	87.1	75.1	74.7	75.5
That is about gender.	81.6	52.9	39.5	79.8

Table 16: Results for supporting hypotheses aimed at detecting texts concerning gender, which in HateCheck corresponds to the categories *transgender people*, and *women*.

hypothesis	accuracy (%)	↓ F_1 (%)	recall (%)	precision (%)
That text is about myself.	97.4	38.5	37.0	40.0
This text is about myself.	96.4	33.7	42.0	28.1
That is about myself.	97.3	33.3	30.9	36.2
This is about myself.	97.0	31.3	30.9	31.6
That example is about myself.	96.2	31.2	39.5	25.8
This example is about myself.	95.9	31.1	42.0	24.6
This text is about us.	85.1	16.8	69.1	9.6
That text is about us.	89.4	16.2	46.9	9.8
That is about us.	88.3	14.8	46.9	8.8
This example is about us.	77.4	12.8	76.5	7.0
That example is about us.	75.8	12.1	76.5	6.6
This is about us.	73.9	10.3	69.1	5.6

Table 17: Results for supporting hypotheses aimed detecting if a text is self-directed.

hypothesis	accuracy (%)
This text supports [X].	100.0
This supports [X].	100.0
That supports [X].	100.0
This example supports [X].	91.9
That example supports [X].	91.9
That text supports [X].	85.5
This text is for [X].	69.4
This is for [X].	50.9
That is for [X].	46.8
That text is for [X].	38.7
This example is for [X].	18.5
That example is for [X].	0.0

Table 18: Results for supporting hypotheses aimed at detecting the stance of an outer text p_1 towards its inner, quoted text p_0. Precision, recall and F_1-score are omitted, since with only positive test examples no false positives and true negatives are possible.