# Using Pre-Trained Language Models for Producing Counter Narratives Against Hate Speech: a Comparative Study

**Serra Sinem Tekiroğlu**[2], **Helena Bonaldi**[1,2], **Margherita Fanton**[1,2*], **Marco Guerini**[2]
[1]University of Trento, Italy
[2]Fondazione Bruno Kessler, Via Sommarive 18, Povo, Trento, Italy
tekiroglu@fbk.eu, hbonaldi@fbk.eu,
margherita.fanton@ims.uni-stuttgart.de, guerini@fbk.eu

## Abstract

In this work, we present an extensive study on the use of pre-trained language models for the task of automatic Counter Narrative (CN) generation to fight online hate speech in English. We first present a comparative study to determine whether there is a particular Language Model (or class of LMs) and a particular decoding mechanism that are the most appropriate to generate CNs. Findings show that autoregressive models combined with stochastic decodings are the most promising. We then investigate how an LM performs in generating a CN with regard to an unseen target of hate. We find out that a key element for successful 'out of target' experiments is not an overall similarity with the training data but the presence of a specific subset of training data, i. e. a target that shares some commonalities with the test target that can be defined *a-priori*. We finally introduce the idea of a pipeline based on the addition of an automatic post-editing step to refine generated CNs.

## 1 Introduction

Hate Speech (HS) has found fertile ground in Social Media Platforms. Actions undertaken by such platforms to tackle online hatred consist in identifying possible sources of hate and removing them by means of content deletion, account suspension or shadow-banning. However, these actions are often interpreted and denounced as censorship by the affected users and political groups (Myers West, 2018). For this reason, such restrictions can have the opposite effect of exacerbating the hostility of the haters (Munger, 2017). An alternative strategy, that is looming on the horizon, is based on the use of Counter Narratives. CNs are "all communicative actions aimed at refuting hate speech through thoughtful and cogent reasons, and true and fact-bound arguments" (Schieb and Preuss, 2016). As a de-escalating measure, CNs have been proven to be successful in diminishing hate, while preserving the freedom of speech (Benesch, 2014; Gagliardone et al., 2015). An example of $<HS, CN>$ pair is shown below:

**HS**: Women are basically childlike, they remain this way most of their lives. Soft and emotional. It has devastated our once great patriarchal civilizations.
**CN**: *Without softness and emotions there would be just brutality and cruelty. Not all women are soft and emotional and many men have these characteristics. To perpetuate these socially constructed gender profiles maintains norms which oppress anybody.*

Based on their effectiveness, CNs have started being employed by NGOs to counter online hate. Since for NGO operators it is impossible to manually write responses to all instances of hate, a line of NLP research has recently emerged, focusing on designing systems to automatically generate CN suggestions (Qian et al., 2019; Tekiroğlu et al., 2020; Fanton et al., 2021; Chung et al., 2021a; Zhu and Bhat, 2021). In this study, our main goal is to compare pre-trained language models (LM) and decoding mechanisms in order to understand their pros and cons in generating CNs. Thus, we use various automatic metrics and manual evaluations with expert judgments to assess several LMs, representing the main categories of the model architectures, and decoding methods. We further test the robustness of the fine-tuned LMs in generating CNs for an unseen target. Results show that autoregressive models are in general more suited for the task, and while stochastic decoding mechanisms can generate more novel, diverse, and informative outputs, the deterministic decoding is useful in scenarios where more generic and less novel (yet 'safer') CNs are needed. Furthermore, in out-of-target experiments we find that the similarity of targets (e.g.

[*] Now at the University of Stuttgart, Germany.



*Findings of the Association for Computational Linguistics: ACL 2022*, pages 3099 - 3114

JEWS and MUSLIMS as religious groups) plays a crucial role for the effectiveness of portability to new targets. We finally show a promising research direction of leveraging gold human edits for building an additional automatic post-editing step to correct errors made by LMs during generation. To the best of our knowledge, this is the first study systematically analysing state of the art pre-trained LMs in CN generation.

## 2 Related Work

In this section we first discuss standard approaches to hate countering and studies on CN effectiveness on Social Media Platforms, then the existing CN data collection and generation strategies.

**Hate countering.** NLP has started addressing the phenomenon of the proliferation of HS by creating datasets for automatic detection (Mathew et al., 2021; Cao et al., 2020; Kumar et al., 2018; Hosseinmardi et al., 2015; Waseem, 2016; Burnap and Williams, 2016). Several surveys provide a review on the existing approaches on the topic (Poletto et al., 2020; Schmidt and Wiegand, 2017; Fortuna and Nunes, 2018), also addressing the ethical challenges of the task (Kiritchenko et al., 2021). Still, automatic detection of HS presents some drawbacks (Vidgen and Derczynski, 2020). First of all, the datasets might include biases, and the models tend to replicate such biases (Binns et al., 2017; Davidson et al., 2019; Sap et al., 2019; Tsvetkov, 2020). Moreover, the end goals for which HS detection is employed are often charged with censorship of the freedom of speech by concerned users (Munger, 2017; Myers West, 2018). In this scenario, NGOs have started employing CNs to counter online hate. CNs have been shown to be effective in reducing linguistic violence (Benesch, 2014; Gagliardone et al., 2015; Schieb and Preuss, 2016; Silverman et al., 2016; Mathew et al., 2019); moreover, even if they might not influence the view of extremists, they are still effective in presenting alternative and non-hateful viewpoints to bystanders (Allison and Bussey, 2016; Anderson et al., 2014).

**CN data collection.** The existing studies for collecting CN datasets employ four main approaches. *Crawling* consists in automatically scraping websites, starting from an HS content and searching for possible CNs among the responses (Mathew et al., 2018, 2019). With *crowdsourcing* CNs are written by non-expert paid workers as responses to provided hate content (Qian et al., 2019). *Nichesourcing* relies on a niche group of experts for data collection (De Boer et al., 2012), and it was employed by Chung et al. (2019) for CN collection using NGO's operators. *Hybrid approaches* use a combination of LMs and humans to collect data (Wallace et al., 2019; Dinan et al., 2019; Vidgen et al., 2020). Studies on CN collection are presented in more detail by Tekiroğlu et al. (2020); Fanton et al. (2021).

**CN generation.** Neural approaches to automatically generate CNs are beginning to be investigated. Fanton et al. (2021); Tekiroğlu et al. (2020); Qian et al. (2019) employ a mix of automatic and human intervention to generate CNs. Zhu and Bhat (2021) propose an entirely automated pipeline of candidate CN generation and filtering. Other lines of work include CN generation for under-resourced languages such as for Italian (Chung et al., 2020), and the generation of knowledge-bound CNs, which allows the production of CNs based on grounded and up-to-date facts and plausible arguments, avoiding the hallucination phenomena (Chung et al., 2021a). Instead, in our work we take a more foundational perspective, which is relevant for all the LM-based pipelines described above. Therefore, we compare and assess various state of the art pre-trained LMs in an end-to-end setting, which is developed as a downstream task for CN generation.

## 3 Methodology

In this section, we present the CN dataset, the language models, and the decoding mechanisms employed for our experiments.

### 3.1 Dataset for fine-tuning

For this study we rely on the dataset proposed by Fanton et al. (2021), which is the only available dataset that grants both the target diversity and the CN quality we aim for. The dataset was collected with a human-in-the-loop approach, by employing an autoregressive LM (GPT-2) paired with three expert human reviewers. It features 5003 $<HS, CN>$ pairs, covering several targets of hate including DISABLED, JEWS, LGBT+, MIGRANTS, MUSLIMS, POC, WOMEN. The residual categories are collapsed to the label OTHER. We partitioned the dataset into training, validation, and test sets with the ratio: $8:1:1$ (i. e. 4003, 500 and 500 pairs), ensuring that all sets share the same

target distribution, and no repetition of HS across the sets is allowed.

### 3.2 Models

We experiment with 5 Transformer based LMs (Vaswani et al., 2017) representing the main categories of the model mechanisms: autoregressive, autoencoder, and seq2seq.

**BERT.** The Bidirectional Encoder Representations from Transformers was introduced by Devlin et al. (2019). It is a bidirectional autoencoder that can be adapted to text generation (Wang and Cho, 2019).

**GPT-2.** The Generative Pre-trained Transformer 2 is an autoregressive model built for text generation (Radford et al., 2019).

**DialoGPT.** The Dialogue Generative Pretrained Transformer is the extension of GPT-2 specifically created for conversational response generation (Zhang et al., 2020).

**BART.** BART is a denoising autoencoder for pretraining seq2seq models (Lewis et al., 2020). The encoder-decoder architecture of BART is composed of a bidirectional encoder and an autoregressive decoder.

**T5.** The Text-to-Text Transfer Transformer proposed by Raffel et al. (2020) is a seq2seq model with an encoder-decoder Transformer architecture.

While all the other models could be fine-tuned directly for the generation task, for BERT we warm-started an encoder-decoder model using BERT checkpoints similar to the BERT2BERT model defined by (Rothe et al., 2020). The fine-tuning details and hyperparameter settings can be found in Appendix A.1.

### 3.3 Decoding mechanisms

We utilize 4 decoding mechanisms: a deterministic (Beam Search) and three stochastic (Top-$k$, Top-$p$, and a combination of the two).

**Beam Search (BS).** The Beam Search algorithm is designed to pick the most-likely sequence (Li et al., 2016; Wiseman et al., 2017).

**Top-$k$ (Top$_k$).** The sampling procedure proposed by Fan et al. (2018) selects a random word from the $k$ most probable ones, at each time step.

**Top-$p$ (Top$_p$).** Also known as Nucleus Sampling, the parameter $p$ indicates the total probability for the pooled candidates, at each time step (Holtzman et al., 2020).

**Combining Top-$p$ and Top-$k$ (Top$_{pk}$).** At decoding stage, it is possible to combine the parameters $p$ and $k$. This is a Nucleus Sampling constrained to the Top-$k$ most probable words.

In our experiments we used the following parameters as default: Beam-Search with 5 beams and repetition penalty = 2; Top-$k$ with $k = 40$; Top-$p$ with $p = .92$; Top$_{pk}$ with $k = 40$ and $p = .92$.

## 4 Evaluation metrics

We use several metrics to evaluate various aspects of the CN generation.

**Overlap Metrics.** These metrics depend on the $n$-gram similarity of the generated outputs to a set of reference texts in order to assess the quality. We used our gold CNs as *references* and the CNs generated by the different models, as *candidates*. In particular, we employed three BLEU variants: BLEU-1 (B-1), BLEU-3 (B-3) and BLEU-4 (B-4) (Papineni et al., 2002), and ROUGE-L (ROU) (Lin, 2004).

**Diversity metrics.** They are used to measure how diverse and novel the produced CNs are. In particular, we utilized *Repetition Rate* (RR) to measure the repetitiveness across generated CNs, in terms of the average ratios of non-singleton $n$-grams present in the corpus (Bertoldi et al., 2013). It should be noted that RR is calculated as a corpus-based repetition score , i.e. inter-CN, instead of calculating intra-CN repetition of $n$-grams only. We also used *Novelty* (NOV) (Wang and Wan, 2018), based on Jaccard similarity, to compute the amount of novel content that is present in the generated CNs as compared to the training data.

**Human evaluation metrics.** Albeit more difficult to attain, human judgments provide a more reliable evaluation and a deeper understanding than automatic metrics (Belz and Reiter, 2006; Novikova et al., 2017). To this end, we specified the following dimensions for the evaluation of CNs. *Suitableness* (SUI): measures how suitable a CN is to the HS in terms of semantic relatedness and in terms of adherence to CN guidelines[1]; *Grammaticality* (GRM): how grammatically correct a generated CN is; *Specificity* (SPE): how specific are the arguments brought by the CN in response to the HS; *Choose-or-not* (CHO): determines whether the annotators would select that CN to post-edit and use it in a real case scenario as in the set up presented by Chung et al. (2021b); *Is-best* (BEST): whether the CN is the absolute best among the ones generated

[1]See for example https://getthetrollsout.org/stoppinghate

for an HS (i. e. whether the annotators would pick up exactly that CN if they had to use it in a real case scenario).

The first three dimensions are rated with a 5-points Likert scale and follow the evaluation procedure described by Chung et al. (2020), whereas both choose-or-not and is-best are binary ratings (0, 1). Choose-or-not allows for multiple CNs to be selected for the same HS, while only one CN can be selected for is-best for each HS.

**Toxicity.**[2] It determines how "rude, disrespectful, or unreasonable" a text is. Toxicity has been employed both to detect the bias present in LMs (Gehman et al., 2020) and as a solution to mitigate such bias (Gehman et al., 2020; Xu et al., 2020).

**Syntactic metrics.** A high syntactic complexity can be used as a proxy for an LM's ability of generating complex arguments. We used the syntactic dependency parser of spaCy[3] For the task, focusing on the following measures: *Maximum Syntactic Depth* (MSD): the maximum depth among the dependency trees calculated over each sentence composing a CN. *Average Syntactic Depth* (ASD): the average depth of the sentences in each CN. *Number of Sentences* (NST): the number of sentences composing a CN.

## 5 Experiments

We performed two sets of experiments: first, we assessed how LMs perform in the task of generating CNs with different decoding mechanisms. Then, we selected the best model from the first round of experiments and tested its generalization capabilities when confronted with an unseen target of hate.

### 5.1 LMs and decoding experiments

For the first round of experiments, in order to avoid possible unfair assessments given by the open nature of the generative task (i. e. a highly suitable CN candidate could be scored low due to its difference from the single reference/gold CN), at test time we allowed the generation of several candidates for each HS+LM+decoding mechanism combination. We loosely drew inspiration from the Rank-$N$ Accuracy procedure and the 'generate, prune, select' procedure (Zhu and Bhat, 2021). In particular, given an LM and a decoding mechanism, we generated 5 CNs for each HS in the test set.

**Automated evaluation and selection** We set up the automatic evaluation strategy as displayed in Figure 1. First, we scored each CN with the overlap metrics presented in Section 4, using the gold CN as a reference. Next, we ranked the candidate CNs with respect to the overlap scores and computed the mean of the rankings. Then, we selected the *best* ones according to the following criteria:

**Best$_{\text{LM}}$** selects the single best CN for an HS among the 20 generated by the 4 models.

**Best$_{\text{D}}$** selects the single best CN for an HS among the 25 generated by the 5 decoding configurations.

**Best$_{\text{LM+D}}$** selects the single best CN among the 5 generated with each model-decoding combination.

Moreover, we assessed the overall corpus-wise quality of the generated CNs with respect to the models, to the decoding mechanisms, and to the model-decoding combinations via the diversity metrics.

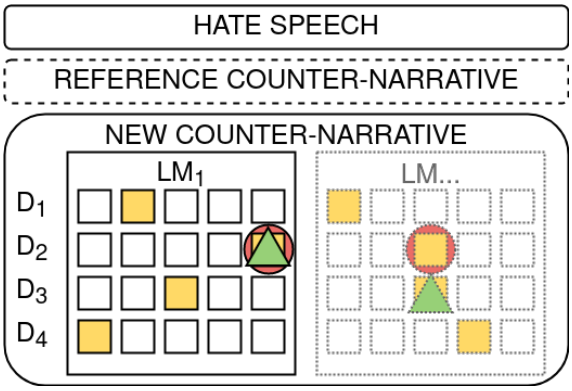


Figure 1: Given an HS, 5 CNs are generated for each model-decoding combination. ● indicates the best CN per model (∈ Best$_{\text{LM}}$). ▲ indicates the best CN per decoding (∈ Best$_{\text{D}}$). ■ indicates the best CN per model-decoding combination (∈ Best$_{\text{LM+D}}$).

**Human evaluation on a sample** To perform the human evaluation we referred to the Best$_{\text{LM}}$ generations and sampled 200 instances from it. Each instance comprises an HS and 5 relevant CNs, each generated by a different model. We recruited 2 annotators who were trained extensively for the task following the procedure used by Fanton et al. (2021). The expert annotators were asked to evaluate the 5 CNs corresponding to the HS, according to the dimensions described in Section 4. We en-

[2] https://www.perspectiveapi.com
[3] https://spacy.io/usage/linguistic-features#dependency-parse

riched the evaluation of this subset with the toxicity and the syntactic metrics.

### 5.2 Results of the first set of experiments

The results of the experiments on the LMs and the decoding mechanisms are reported in this section[4].

**Best Model** The results of the comparison of the models on the $\text{Best}_{\text{LM}}$ generations can be found in Table 1. Regarding the overlap and diversity metrics, DialoGPT records the best or the second best score in all the metrics, apart from novelty where it still achieves a high score (0.643) close to the best performance (0.655). T5 also achieves high scores, especially on ROUGE, BLEU-1 and novelty.

BART, instead, is the best model according to human evaluation metrics, apart from specificity. On the other hand, it shows poor performances in terms of diversity metrics, indicating that it tends to produce grammatical and suitable but very generic responses.

BERT records the worst scores for all the overlap and diversity metrics apart from novelty. However, it also achieves the best syntactic metric results. Therefore, it is evident that BERT's output is more complex, but very repetitive. The combination of these aspects eventually affects the clarity of BERT's output such that it yields poor results in the human evaluation, in particular for grammaticality (4.2, while other models are above 4.6). This poor grammaticality can also explain the syntactic scores since the spaCy dependency parser was not trained to handle ungrammatical text and this could actually inflates the ASD and MSD scores.

GPT-2 overall yields very competitive results for several groups of metrics. It obtains the second-highest novelty score (0.653) and the best RR (7.736). It also achieves the second best results on BLEU-3, maximum syntactic depth and number of sentences, and the best results on toxicity and specificity (2.880) indicating the ability to produce complex, suitable, focused and diverse CNs.

After the human evaluation we ran a qualitative interview with the annotators, whose feedback on the data strengthened the results we observed and the conclusion we drew. For instance, they reported the repetition of simple, yet catch-them-all, expressions (e.g. "they are our brothers and sisters") regardless of the target. Further inspections found that those CNs were mainly produced by BERT, which is in line with BERT's RR score.

**Best Decoding mechanism.** The results calculated on $\text{Best}_{\text{D}}$ output are presented in Table 2. $\text{Top}_k$ is the best performing decoding mechanism achieving the best results on the diversity metrics, BLEU-3 and BLEU-4. It is also the best performing for specificity, maximum syntactic depth and number of sentences, and the second best for average syntactic depth and toxicity.

The other stochastic decoding mechanisms perform well too. $\text{Top}_p$ yields competitive results on both diversity and overlap metrics; it is the second best for specificity, and achieves good results on the syntactic metrics. $\text{Top}_{pk}$ has a good performance on the overlap metrics. It obtains the second-highest scores in most of the human evaluation metrics and the lowest in toxicity, and it reaches a reasonable specificity score.

On the other hand, BS does not achieve particularly good results, except for the ROUGE score. Even if it is the best decoding with respect to the human evaluation, this comes at the cost of specificity and diversity. Through a post-hoc manual analysis we observed that it was due to the deterministic nature of BS, that tends to choose the most probable sequences, i. e. the "safest", thus resulting in vague and repetitive outputs.

**Best Model-Decoding combination** Here we briefly discuss the results of the evaluation obtained on the $\text{Best}_{\text{LM+D}}$ generations. In particular, the autoregressive models GPT-2 and DialoGPT behave similarly with similar decoding mechanisms, such that BS outputs the best results for almost all the overlap metrics, and the worst for the diversity metrics. On the contrary, for the other models, the results achieved with stochastic decoding mechanisms are the best for the overlap metrics. In almost all the cases, we observe that the stochastic decoding mechanisms perform better on syntactic and diversity metrics and on toxicity, while for the human evaluation metrics BS tends to be the best, except for specificity. A detailed discussion can be found in Appendix A.2.

**Discussion.** In this set of experiments, we found that the autoregressive models perform the best according to a combination of several metrics that we deem particularly relevant (e.g. more novel, diverse, and informative outputs). Of course more repetitive and conservative outputs can be preferred

[4]The training details for all the models we employed are described in Appendix A.1

| | Overlap | | | | Diversity | | Toxicity | Syntactic metrics | | | Human evaluation | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | **ROU** | **B-1** | **B-3** | **B-4** | **RR** | **NOV** | **-** | **ASD** | **MSD** | **NST** | **SUI** | **SPE** | **GRM** | **CHO** | **BEST** |
| BART | 0.268 | 0.277 | 0.085 | **0.051** | 20.722 | 0.560 | 0.420 | 4.311 | 4.965 | 1.740 | **3.790** | 2.552 | **4.937** | **0.840** | **0.272** |
| BERT | 0.237 | 0.277 | 0.073 | 0.037 | 24.747 | 0.605 | 0.406 | **5.008** | **6.160** | **2.280** | 3.135 | 2.647 | 4.247 | 0.717 | 0.122 |
| T5 | **0.274** | **0.302** | 0.083 | 0.042 | 8.548 | **0.655** | 0.359 | **4.692** | 5.325 | 1.715 | 2.872 | 2.402 | 4.680 | 0.642 | 0.090 |
| DialoGPT | **0.273** | **0.304** | **0.093** | **0.052** | **8.248** | 0.643 | **0.343** | 4.677 | 5.575 | 1.895 | 3.392 | **2.755** | **4.880** | 0.767 | 0.245 |
| GPT-2 | 0.264 | 0.297 | **0.088** | 0.050 | **7.736** | **0.653** | **0.342** | 4.584 | **5.595** | **2.240** | **3.555** | **2.880** | 4.867 | **0.795** | **0.270** |

Table 1: Results of the overlap and diversity metrics are calculated on the $\text{Best}_{\text{LM}}$ generations while the toxicity, the syntactic metrics and the human evaluation are calculated on the corresponding subset.

| | Overlap | | | | Diversity | | Toxicity | Syntactic metrics | | | Human evaluation | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | **ROU** | **B-1** | **B-3** | **B-4** | **RR** | **NOV** | **-** | **ASD** | **MSD** | **NST** | **SUI** | **SPE** | **GRM** | **CHO** | **BEST** | **n** |
| BS | **0.287** | 0.299 | 0.096 | 0.059 | 21.579 | 0.561 | 0.398 | 4.415 | 5.048 | 1.684 | **3.936** | 2.497 | **4.925** | **0.826** | **0.222** | %18.7 |
| $\text{Top}_{pk}$ | **0.287** | **0.320** | **0.106** | 0.059 | 11.404 | 0.639 | **0.352** | 4.676 | 5.488 | 1.932 | **3.324** | 2.647 | **4.688** | **0.764** | **0.212** | %29.3 |
| $\text{Top}_{k}$ | 0.282 | 0.314 | **0.106** | **0.060** | **10.076** | **0.652** | **0.374** | **4.704** | **5.756** | **2.133** | 3.155 | **2.716** | 4.659 | 0.716 | 0.183 | %27.1 |
| $\text{Top}_{p}$ | 0.285 | **0.319** | 0.105 | **0.060** | **11.270** | **0.640** | 0.381 | **4.753** | **5.671** | **2.068** | 3.149 | **2.687** | 4.681 | 0.723 | 0.189 | %24.9 |

Table 2: The results for the overlap and diversity metrics are calculated on the $\text{Best}_{\text{D}}$ generations: for each decoding mechanism, there are 2500 CNs. The remaining metrics are calculated on a subset of 1000 CNs: the distribution of which is shown in the column "n".

when high precision of suitable CNs are required at the expense of being more generic and less novel. Still, for what concerns autoregressive models it could be argued that the good performance of the GPT-2 model we fine-tuned is due to the fact that generated CNs and gold CNs derive from a similar distribution (GPT-2 was employed in the human-in-the-loop process used to create the reference dataset from Fanton et al. (2021)). While we recognize that this could partially explain the performance of our GPT-2 model, it does not explain the performance of DialoGPT, which is pre-trained on a completely different dataset. Therefore, we can reasonably conclude that autoregressive models are particularly suited for the task, regardless of the pre-training data.

With respect to the decoding mechanisms, we record high repetitiveness and low novelty for the deterministic decoding BS. Even if it reaches high scores in most of the human evaluation metrics, it fails to produce specific CNs ending up in generating suitable, yet generic responses. On the contrary, stochastic decoding mechanisms produce more novel and specific responses.

Example CNs generated in this session of experiments, along with some qualitative analysis, can be found in Appendix A.3.

### 5.3 Leave One Target Out experiments

In the second stage, we built a set of cross-domain experiments to capture the generalization capabilities of the best LM determined in the previous experiments. Specifically, we concentrate on assessing how much a pre-trained language model fine-tuned on a pool of hate targets can generalize to an unseen target.

Thus, for the out of target experiment we selected the LM that we deem the most prominent in order to reduce the number of LM configurations to compare. In particular, since we want to examine the generalization capability of the LM, the generation of *novel* CNs, in comparison to the training data, is given primary importance. Secondly, *specificity* is also crucial since it signifies the ability of the LM/decoding mechanism in generating accurate CNs and avoiding vague yet suitable, catch-all CNs. In contrast, repetitiveness is an undesirable feature of CNs, as it signals the tendency of a model to produce less flexible content. Given these considerations, we chose to employ GPT-2 with $\text{Top}_{k}$ decoding for the Leave One Target Out (LOTO) experiments since it is the configuration achieving the best trade-off amongst all the others.

This set of experiments is structured in 3 steps, replicated for each of the selected targets. We selected the targets with the highest number of examples (MUSLIMS, MIGRANTS, WOMEN, LGBT+ and JEWS) to have a sufficient sized test set for each configuration.

First, we sampled from the Fanton et al. (2021) dataset 600 pairs for each LOTO target, in order to have a balanced setting. Additionally, POC and DISABLED were always kept in the training set, and we removed multi-target cases from OTHER. The resulting dataset consists of 3729 instances (further details are provided in Appendix A.4). Sec-

ondly, we fine-tuned 5 different configurations of the LM, and in each configuration one of the 5 LOTO targets is not present in the training data: LM$_{\text{-JEWS}}$, LM$_{\text{-LGTB+}}$, LM$_{\text{-MIGRANTS}}$, LM$_{\text{-MUSLIMS}}$ and LM$_{\text{-WOMEN}}$. Finally, we tested each LOTO model on the 600 HSs in the test set made of "left out" target examples. For instance, the model LM$_{\text{-JEWS}}$ is used for generating the CNs for the target JEWS, after being trained on $<HS, CN>$ data without any instances with the label JEWS. We generated 5 CNs for each HS and selected the best CN according to the procedure described in Section 5.1.

**Results of LOTO experiments**

We analyse the CNs generated with the LOTO models through overlap and diversity metrics (Table 3). We refer to Appendix A.4 for the comparison between RR calculated on the candidate CNs and the reference CNs of the Fanton et al. (2021) dataset.

For all the targets we record higher novelty scores as compared to the previous experiments. Higher novelty ranges indicate that conditioning with new material (i. e. HS for the unseen targets) induces GPT-2 to produce new arguments. On the other hand, as expected, the overlap scores for LOTO are remarkably lower than those from the previous experiments (Table 3). Therefore, we can infer that generalizing to an unseen target is harder than generalizing to an unseen HS.

| LOTO | Overlap | | | | Diversity | |
|---|---|---|---|---|---|---|
| Target | **ROU** | **B-1** | **B-3** | **B-4** | **RR** | **NOV** |
| JEWS | 0.1609 | 0.1842 | 0.0134 | 0.0035 | 4.796 | 0.718 |
| LGBT+ | 0.1599 | 0.1828 | 0.0149 | 0.0055 | **4.620** | 0.718 |
| MIGRANTS | 0.1659 | 0.1915 | 0.0163 | 0.0038 | 4.707 | **0.720** |
| MUSLIMS | **0.1743** | **0.1934** | **0.0197** | **0.0059** | 5.314 | 0.712 |
| WOMEN | **0.1755** | **0.1988** | **0.0195** | **0.0068** | **4.632** | **0.729** |

Table 3: The overlap and diversity metrics scores for the various LOTO configurations.

We also found out that the CNs generated in the LM$_{\text{-MUSLIMS}}$ and LM$_{\text{-WOMEN}}$ configurations obtain the highest overlap scores (Table 3). We hypothesize that the high scores can be explained by the presence of a target in the LOTO training that is highly similar to the left out one. To this end, we computed the novelty between each target subset of the training data and the LOTO test data for that configuration (see Appendix A.4 for details). The reference CNs for LM$_{\text{-MUSLIMS}}$ record the lowest novelty scores with respect to the JEWS subset of the training set (i. e. 0.761). Thus, it

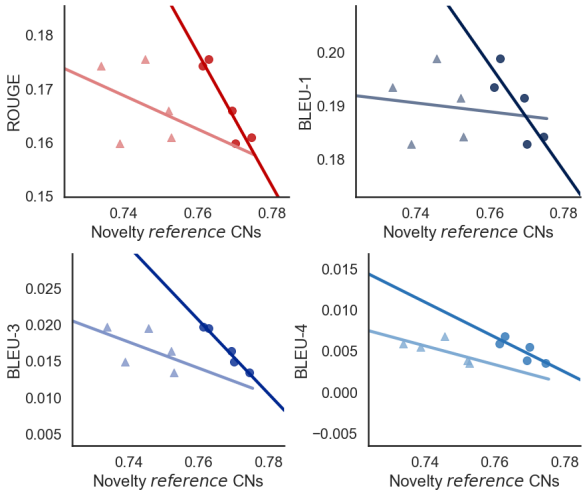


Figure 2: The correlation between the novelty of the reference CNs and overlap metrics: in each plot, the dots and the darker line correspond to the most influential target; the triangles and the lighter line correspond to the results calculated without it.

can be interpreted as the most influential portion of training data for the target MUSLIMS. On the other hand, for LM$_{\text{-WOMEN}}$ the highest influence is recorded with the LGBT+ subset of the training data (i. e. 0.763). These results can be explained by the semantic similarity of the target MUSLIMS to JEWS, both being religious groups; and of WOMEN to LGBT+, both being related to gender issues.

As a complementary analysis, we consider two different computations of the reference CN novelty: with respect to the most influential target for each LOTO configuration, and with respect to the LOTO training data without the most influential target. We computed the Pearson correlation between the overlap metrics and each of the two novelty computations. In Figure 2, we observe that removing the influential target from the training data strongly decreases the correlation with the overlap metrics (from an average of -0.889 to -0.416). Consequently, we can conclude that to obtain high overlap results in the LOTO experiments, it is necessary that the training data contains a target strongly connected to the left out one. Most importantly, this connection is not arbitrarily decided but it is based on an *a-priori* semantic similarity of the targets as exemplified before.

Finally, we chose to generate also with the BS decoding mechanism, to use it as a baseline and compare it to the stochastic decoding mechanism (Top-$k$). In particular, we computed the Pearson correlation between the novelty of the reference

CNs and the novelty of the candidate CNs with respect to the corresponding training data (Figure 3). We can observe that for the BS generation the novelty of the candidate CNs is lower than Top-$k$ (0.67-0.74 vs. 0.75-0.77) and the correlation with the novelty of the reference is weaker (0.53 vs. 0.59). This confirms the lower generalization ability with the deterministic decoding mechanism (as compared to the stochastic) that tends to produce generic and repetitive responses regardless of the semantic distances of the LOTO targets from the training data.

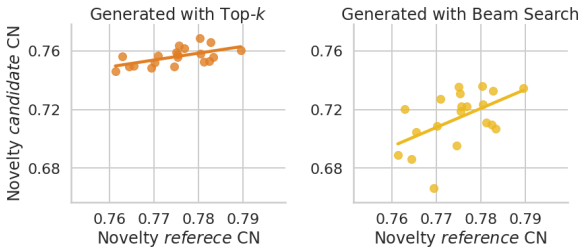


Figure 3: Reference and candidate CNs novelty, for Top-$k$ and BS LOTO generations.

## 6 Automatic Post-Editing

In the previous experiments we fine-tuned our models making resort to $<HS, CN>$ pairs alone. Still the Fanton et al. (2021) dataset contains additional information that can be useful for our task: i. e. the original GPT-2 generation before undergoing human post-editing.

Thus, as a final experiment, we propose to further improve the CN generation by moving from an end-to-end framework to a two stage pipeline, by decoupling CN generation from its 'final refinement'. In particular we propose the adoption of an Automatic Post-Editing (APE) stage in order to capture and utilize the nuances among the machine generated CNs and their human post-edited versions. APE, which is used for automatically correcting errors made by machine translation (MT) systems before performing actual human post-editing, has been an important tool for MT (Knight and Chander, 1994; do Carmo et al., 2021). Considering its effectiveness in MT, we hypothesize that building a pipeline with CN generation and APE could alleviate the requirement of the final manual post-editing (Allen and Hogan, 2000; Chatterjee et al., 2019) to achieve better constructed CNs.

To this end, we fine-tuned another instance of GPT-2 medium model specifically for the post-editing task using the $<HS, CN_{or}, CN_{pe}>$ triplets[5], where $CN_{or}$ and $CN_{pe}$ denote the CNs originally generated by an LM and their human post-edited versions, respectively. The triplets were then filtered by removing those for which $CN_{or} = CN_{pe}$. More details about the experiment settings can be found in Appendix A.5.

| Data | $CN_{ape}$ | $CN_{or}$ | N/A |
|---|---|---|---|
| Fanton et al. (2021) | 26 | 14 | 60 |
| GPT-2 Top$_k$ | 37 | 19 | 44 |

Table 4: The human annotation results for the APE experiments in terms of average preference percentages.

We have conducted two human evaluations over the subsets of: i) the $CN_{or}$ of the Fanton et al. (2021) test samples, ii) the CN outputs of the best model and decoding mechanism combination provided as the results of the first set of experiments, that yielded the top 50 Translation Error Rate (TER) (Snover et al., 2006) scores with respect to the $CN_{or}$s. The two expert annotators were asked to state their preferences among the 2 randomly sorted CNs, $CN_{or}$ and $CN_{ape}$ (automatically post-edited output), for a given HS. The annotators were also allowed to decide on a tie. Results, shown in Table 4, indicate that, albeit there are often ties and only a subset of $CN_{or}$ is actually modified, when there is a preference, it is predominantly in favour of the automatically post-edited versions over the GPT-2 generated CNs (26% vs. 14% for the test set, and 37% vs. 19% for the GPT-2 Top$_k$ generations, on average). Regarding the experiment results, we believe that APE is a highly promising direction to increase the efficacy of the CN generation models where generation quality and diversity is crucial, and considering that obtaining/enlarging expert datasets to train better models is not simple.

## 7 Conclusion

In this work, we focus on automatic CN generation as a downstream task. First, we present a comparative study to determine the performances and peculiarities of several pre-trained LMs and decoding mechanisms. We observe that the best results (in term of novelty and specificity) overall are achieved

[5]This is in line with the APE paradigm where the triplet is made of $<$*source sentece*, $MT_{output}$, *human post-edits*$>$.

by the autoregressive models with stochastic decoding: GPT-2 with the Top$_k$ decoding mechanism, and DialoGPT with the combination Top$_{pk}$. At the same time deterministic decoding can be used when more generic yet 'safer' CNs are preferred.

Then, we investigate the performances of LMs in zero-shot generation for unseen targets of hate. Hence, we fine-tuned 5 different versions of GPT-2, leaving out the examples pertaining to one target at each turn. We find out that for each configuration/version, there is a subset of the training data which is more influential with respect to the generated data (i. e. a target that shares some commonalities with the test target that can be defined a-priori). Finally, we introduce an experiment by training an automatic post-editing module to further improve the CN generation quality. The notable human evaluation results paves the way for a promising future direction that decouples CN generation from its 'final refinement'.

## Ethical Considerations

Although tackling online hatred through CNs inherently protects the freedom of speech and has been proposed as a better alternative to the detect-remove-ban approaches, automatization of CN generation can still raise some ethical concerns and some measures must be taken to avoid undesired effects during research. Thus, we address the relevant ethical considerations and our remedies as follows:

**Annotation Guidelines:** The well-being of the annotators was our top priority during the whole study. Therefore, we strictly followed the guidelines created for CN studies (Fanton et al., 2021) that were adapted from (Vidgen et al., 2019). The human evaluations have been conducted with the help of two expert annotators in CNs. These experts were already trained for the CN generation task and employed for the work presented by (Fanton et al., 2021). We further instructed them in the aims of each experiment, clearly explained the evaluation tasks, and then we exemplified proper evaluation of $<HS, CN>$ pairs using various types of CNs. Most importantly, we limited the exposure to hateful content by providing a daily time limit of annotation. Concerning the demographics, due to the harmful content that can be found in the data, all annotators were adult volunteers, perfectly aware of the objective of the study.

**Dataset.** We purposefully chose an expert-based dataset in order to avoid the risk of modeling the language of real individuals to (i) prevent any privacy issue, (ii) avoid to model inappropriate CNs (e.g. containing abusive language) that could be produced by non-experts. The dataset also focuses on the CN diversity while keeping the HSs as stereotypical as possible so that our CN generation models have a very limited diversity on the hateful language, nearly precluding the misuse.

**Computational Task.** CN generation models are not meant to be used in an autonomous way, since even the best models can still produce substandard CNs containing inappropriate or negative language. Instead, following a Human–computer cooperation paradigm, our focus is on building models that can be helpful to NGO operators by providing them diverse and novel CN candidates for their hate countering activities and speed up the manual CN writing to a certain extent. This approach also gives ground to some of the measures we used during evaluation (namely choose-or-not and is-best).

**Model Distribution.** In addition to the limited and simplified hateful content in the dataset we selected, we further reduce the risk of misuse by choosing a specific distribution strategy: i) we only make available the non-autoregressive models in order to eliminate the risk of using over-generation for hate speech creation, ii) we distribute such models only for research purposes and through a request based procedure in order to keep track of the possible users.

## References


Jeffrey Allen and Christopher Hogan. 2000. Toward the development of a post editing module for raw machine translation output: A controlled language perspective. In *Third International Controlled Language Applications Workshop (CLAW-00)*, pages 62–71.

Kimberley R Allison and Kay Bussey. 2016. Cyberbystanding in context: A review of the literature on witnesses' responses to cyberbullying. *Children and Youth Services Review*, 65:183–194.

Jenn Anderson, Mary Bresnahan, and Catherine Musatics. 2014. Combating weight-based cyberbullying on facebook with the dissenter effect. *Cyberpsychology, Behavior, and Social Networking*, 17(5):281–286.

Anja Belz and Ehud Reiter. 2006. Comparing automatic and human evaluation of nlg systems. In *11th*

*conference of the european chapter of the association for computational linguistics*, pages 313–320.

Susan Benesch. 2014. Countering dangerous speech: New ideas for genocide prevention. *Washington, DC: United States Holocaust Memorial Museum*.

Nicola Bertoldi, Mauro Cettolo, and Marcello Federico. 2013. Cache-based online adaptation for machine translation enhanced computer assisted translation. In *MT-Summit*, pages 35–42.

Reuben Binns, Michael Veale, Max Van Kleek, and Nigel Shadbolt. 2017. Like trainer, like bot? inheritance of bias in algorithmic content moderation. In *Social Informatics*, pages 405–415, Cham. Springer International Publishing.

Pete Burnap and Matthew L Williams. 2016. Us and them: identifying cyber hate on twitter across multiple protected characteristics. *EPJ Data Science*, 5(1):11.

Rui Cao, Roy Ka-Wei Lee, and Tuan-Anh Hoang. 2020. Deephate: Hate speech detection via multi-faceted text representations. In *12th ACM Conference on Web Science*, pages 11–20.

Félix do Carmo, Dimitar Shterionov, Joss Moorkens, Joachim Wagner, Murhaf Hossari, Eric Paquin, Dag Schmidtke, Declan Groves, and Andy Way. 2021. A review of the state-of-the-art in automatic post-editing. *Machine Translation*, 35(2):101–143.

Rajen Chatterjee, Christian Federmann, Matteo Negri, and Marco Turchi. 2019. Findings of the wmt 2019 shared task on automatic post-editing. In *Proceedings of the Fourth Conference on Machine Translation (Volume 3: Shared Task Papers, Day 2)*, pages 11–28.

Yi-Ling Chung, Elizaveta Kuzmenko, Serra Sinem Tekiroğlu, and Marco Guerini. 2019. CONAN - COunter NArratives through nichesourcing: a multilingual dataset of responses to fight online hate speech. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2819–2829, Florence, Italy. Association for Computational Linguistics.

Yi-Ling Chung, Serra Sinem Tekiroğlu, and Marco Guerini. 2020. Italian counter narrative generation to fight online hate speech. In *Proceedings of the Seventh Italian Conference on Computational Linguistics CLiC-it*.

Yi-Ling Chung, Serra Sinem Tekiroğlu, and Marco Guerini. 2021a. Towards knowledge-grounded counter narrative generation for hate speech. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 899–914, Online. Association for Computational Linguistics.

Yi-Ling Chung, Serra Sinem Tekiroğlu, Sara Tonelli, and Marco Guerini. 2021b. Empowering ngos in countering online hate messages. *Online Social Networks and Media*, 24:100150.

Thomas Davidson, Debasmita Bhattacharya, and Ingmar Weber. 2019. Racial bias in hate speech and abusive language detection datasets. In *Proceedings of the Third Workshop on Abusive Language Online*, pages 25–35.

Victor De Boer, Michiel Hildebrand, Lora Aroyo, Pieter De Leenheer, Chris Dijkshoorn, Binyam Tesfa, and Guus Schreiber. 2012. Nichesourcing: harnessing the power of crowds of experts. In *International Conference on Knowledge Engineering and Knowledge Management*, pages 16–20. Springer.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186.

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

Angela Fan, Mike Lewis, and Yann Dauphin. 2018. Hierarchical neural story generation. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 889–898, Melbourne, Australia. Association for Computational Linguistics.

Margherita Fanton, Helena Bonaldi, Serra Sinem Tekiroğlu, and Marco Guerini. 2021. Human-in-the-loop for data collection: a multi-target counter narrative dataset to fight online hate speech. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics*, Online. Association for Computational Linguistics.

Paula Fortuna and Sérgio Nunes. 2018. A survey on automatic detection of hate speech in text. volume 51, page 85. ACM.

Iginio Gagliardone, Danit Gal, Thiago Alves, and Gabriela Martinez. 2015. *Countering online hate speech*. Unesco Publishing.

Samuel Gehman, Suchin Gururangan, Maarten Sap, Yejin Choi, and Noah A Smith. 2020. Realtoxicityprompts: Evaluating neural toxic degeneration in language models. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: Findings*, pages 3356–3369.

Ari Holtzman, Jan Buys, Li Du, Maxwell Forbes, and Yejin Choi. 2020. The curious case of neural text degeneration. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*.

Homa Hosseinmardi, Sabrina Arredondo Mattson, Rahat Ibn Rafiq, Richard Han, Qin Lv, and Shivakant Mishra. 2015. Detection of cyberbullying incidents on the instagram social network. *arXiv preprint arXiv:1503.03909*.

Svetlana Kiritchenko, Isar Nejadgholi, and Kathleen C Fraser. 2021. Confronting abusive language online: A survey from the ethical and human rights perspective. *Journal of Artificial Intelligence Research*, 71:431–478.

Kevin Knight and Ishwar Chander. 1994. Automated postediting of documents. In *AAAI*, volume 94, pages 779–784.

Ritesh Kumar, Atul Kr Ojha, Shervin Malmasi, and Marcos Zampieri. 2018. Benchmarking aggression identification in social media. In *Proceedings of the First Workshop on Trolling, Aggression and Cyberbullying (TRAC-2018)*, pages 1–11.

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.

Jiwei Li, Will Monroe, Alan Ritter, Dan Jurafsky, Michel Galley, and Jianfeng Gao. 2016. Deep reinforcement learning for dialogue generation. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1192–1202, Austin, Texas. Association for Computational Linguistics.

Chin-Yew Lin. 2004. Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out*, pages 74–81.

Binny Mathew, Navish Kumar, Pawan Goyal, Animesh Mukherjee, et al. 2018. Analyzing the hate and counter speech accounts on twitter. *arXiv preprint arXiv:1812.02712*.

Binny Mathew, Punyajoy Saha, Hardik Tharad, Subham Rajgaria, Prajwal Singhania, Suman Kalyan Maity, Pawan Goyal, and Animesh Mukherjee. 2019. Thou shalt not hate: Countering online hate speech. In *Proceedings of the International AAAI Conference on Web and Social Media*, volume 13, pages 369–380.

Binny Mathew, Punyajoy Saha, Seid Muhie Yimam, Chris Biemann, Pawan Goyal, and Animesh Mukherjee. 2021. Hatexplain: A benchmark dataset for explainable hate speech detection. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 14867–14875.

Kevin Munger. 2017. Tweetment effects on the tweeted: Experimentally reducing racist harassment. *Political Behavior*, 39(3):629–649.

Sarah Myers West. 2018. Censored, suspended, shadowbanned: User interpretations of content moderation on social media platforms. *New Media & Society*, 20(11):4366–4383.

Jekaterina Novikova, Ondrej Dusek, Amanda Cercas Curry, and Verena Rieser. 2017. Why we need new evaluation metrics for nlg. In *2017 Conference on Empirical Methods in Natural Language Processing*, pages 2231–2242. Association for Computational Linguistics.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting on association for computational linguistics*, pages 311–318. Association for Computational Linguistics.

Fabio Poletto, Valerio Basile, Manuela Sanguinetti, Cristina Bosco, and Viviana Patti. 2020. Resources and benchmark corpora for hate speech detection: a systematic review. *Language Resources and Evaluation*, pages 1–47.

Jing Qian, Anna Bethke, Yinyin Liu, Elizabeth Belding, and William Yang Wang. 2019. A benchmark dataset for learning to intervene in online hate speech. 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 4757–4766, Hong Kong, China. Association for Computational Linguistics.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners. *OpenAI Blog*, 1(8).

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.

Sascha Rothe, Shashi Narayan, and Aliaksei Severyn. 2020. Leveraging pre-trained checkpoints for sequence generation tasks. *Transactions of the Association for Computational Linguistics*, 8:264–280.

Maarten Sap, Dallas Card, Saadia Gabriel, Yejin Choi, and Noah A Smith. 2019. The risk of racial bias in hate speech detection. In *Proceedings of the 57th annual meeting of the association for computational linguistics*, pages 1668–1678.

Carla Schieb and Mike Preuss. 2016. Governing hate speech by means of counterspeech on facebook. In *66th ICA Annual Conference, at Fukuoka, Japan*, pages 1–23.

Anna Schmidt and Michael Wiegand. 2017. A survey on hate speech detection using natural language processing. In *Proceedings of the Fifth International*

*Workshop on Natural Language Processing for Social Media*, pages 1–10.

Tanya Silverman, Christopher J Stewart, Jonathan Birdwell, and Zahed Amanullah. 2016. The impact of counter-narratives. *Institute for Strategic Dialogue, London. https://www. strategicdialogue. org/wp-content/uploads/2016/08/Impact-of-Counter-Narratives_ONLINE. pdf–73*.

Matthew Snover, Bonnie Dorr, Richard Schwartz, Linnea Micciulla, and John Makhoul. 2006. A study of translation edit rate with targeted human annotation. In *Proceedings of association for machine translation in the Americas*, volume 200, 6. Cambridge, MA.

Serra Sinem Tekiroğlu, Yi-Ling Chung, and Marco Guerini. 2020. Generating counter narratives against online hate speech: Data and strategies. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1177–1190, Online. Association for Computational Linguistics.

Mengzhou Xia Anjalie Field Yulia Tsvetkov. 2020. Demoting racial bias in hate speech detection. *SocialNLP 2020*, page 7.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008.

Bertie Vidgen and Leon Derczynski. 2020. Directions in abusive language training data, a systematic review: Garbage in, garbage out. *Plos one*, 15(12):e0243300.

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.

Bertie Vidgen, Tristan Thrush, Zeerak Waseem, and Douwe Kiela. 2020. Learning from the worst: Dynamically generated datasets to improve online hate detection. *arXiv preprint arXiv:2012.15761*.

Eric Wallace, Pedro Rodriguez, Shi Feng, Ikuya Yamada, and Jordan Boyd-Graber. 2019. Trick me if you can: Human-in-the-loop generation of adversarial question answering examples. *Transactions of the Association for Computational Linguistics*, 7(0):387–401.

Alex Wang and Kyunghyun Cho. 2019. BERT has a mouth, and it must speak: BERT as a Markov random field language model. In *Proceedings of the Workshop on Methods for Optimizing and Evaluating Neural Language Generation*, pages 30–36, Minneapolis, Minnesota. Association for Computational Linguistics.

Ke Wang and Xiaojun Wan. 2018. Sentigan: Generating sentimental texts via mixture adversarial networks. In *IJCAI*, pages 4446–4452.

Zeerak Waseem. 2016. Are you a racist or am i seeing things? annotator influence on hate speech detection on twitter. In *Proceedings of the first workshop on NLP and computational social science*, pages 138–142.

Sam Wiseman, Stuart Shieber, and Alexander Rush. 2017. Challenges in data-to-document generation. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2253–2263, Copenhagen, Denmark. Association for Computational Linguistics.

Jing Xu, Da Ju, Margaret Li, Y-Lan Boureau, Jason Weston, and Emily Dinan. 2020. Recipes for safety in open-domain chatbots. *arXiv e-prints*, pages arXiv–2010.

Yizhe Zhang, Siqi Sun, Michel Galley, Yen-Chun Chen, Chris Brockett, Xiang Gao, Jianfeng Gao, Jingjing Liu, and Bill Dolan. 2020. DIALOGPT : Large-scale generative pre-training for conversational response generation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 270–278, Online. Association for Computational Linguistics.

Wanzheng Zhu and Suma Bhat. 2021. Generate, prune, select: A pipeline for counterspeech generation against online hate speech. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 134–149.

# A Appendix

## A.1 Fine-tuning details

Table 5 summarizes the details of the training of each model employed in the first session of experiments.

| | BA | EP | PAR | LR | PER | TL | EL |
|---|---|---|---|---|---|---|---|
| BART (base) | 4 | 4 | 139 M | 2E-05 | 24.659 | 2.358 | 2.417 |
| BERT Seq2Seq (base) | 4 | 3 | 247 M | 3E-05 | 11.209 | 2.845 | 3.205 |
| T5 (base) | 2 | 3 | 223 M | 5E-05 | 10.9248 | 2.412 | 3.205 |
| DialoGPT (medium) | 4 | 2 | **355 M** | 5E-05 | **6.085** | **1.425** | **1.806** |
| GPT-2 (medium) | 2 | 2 | **355 M** | 5E-05 | **8.929** | **1.320** | **2.189** |

Table 5: The training details for all the models employed for the first collection of experiment: the batch size (BA), number of training epochs (EP), parameters (PAR), the learning rate (LR), perplexity (PER), training and evaluation loss (TL and EL).

Since LM sizes are very different for each model and since our main focus is not studying performances according to LM dimension growth, as a rule-of-thumb, we chose one version smaller than the large version of each model provided that they all have the same order of magnitude. This corresponds to the *medium* versions for both DialoGPT and GPT-2, and *base* versions for the other models. GPT-2 and DialoGPT achieve the lowest perplexity, training and evaluation loss, thus indicating a slightly more successful fine-tuning, which are reflected in the evaluations throughout the study.

We conducted a hyper-parameter search during the training phase of each model using the search space: learning-rate:$\{1e-5, 2e-5, 3e-5, 4e-5, 5e-5\}$, warm-up ratio:$\{0, 0.1\}$, batch-size:$\{2, 4\}$, epochs:$\{2, 3, 4, 5\}$. It has been conducted using Optuna, with 10 trials, optimized on minimizing the evaluation loss during training.

## A.2 Best models-decoding combination

Here we discuss the results for the overlap and diversity metrics obtained on the $\text{Best}_{\text{LM+D}}$ generations (Table 6), and those calculated on the human evaluation subset (Tables 7 and 8).

**BART.** BART performs well with the stochastic decoding methods, in particular: $\text{Top}_p$ for overlap, diversity, syntactic metrics, and grammaticality; $\text{Top}_k$ for overlap metrics and toxicity, whereas $\text{Top}_{pk}$ is the best decoding approach on human evaluation and RR, and the second best on ROUGE and BLEU-1. On the contrary, BART does not achieve good results with deterministic approaches (i. e. BS).

| | Overlap | | | | Diversity | |
|---|---|---|---|---|---|---|
| | ROU | B-1 | B-3 | B-4 | RR | NOV |
| BART BS | 0.2108 | 0.2129 | 0.0486 | 0.0283 | 21.1102 | **0.5692** |
| BART $\text{Top}_{pk}$ | **0.2331** | **0.2300** | 0.0605 | 0.0365 | **20.2645** | 0.5567 |
| BART $\text{Top}_k$ | **0.2349** | **0.2333** | **0.0652** | **0.0385** | 20.6587 | 0.5575 |
| BART $\text{Top}_p$ | 0.2329 | 0.2300 | **0.0621** | **0.0374** | **20.5476** | **0.5586** |
| BERT BS | 0.1735 | 0.2108 | 0.0249 | 0.0113 | 38.0349 | 0.5864 |
| BERT $\text{Top}_{pk}$ | **0.2034** | 0.2311 | **0.0484** | **0.0231** | **23.4417** | 0.6098 |
| BERT $\text{Top}_k$ | 0.2032 | **0.2320** | 0.0483 | 0.0229 | **22.2546** | **0.6129** |
| BERT $\text{Top}_p$ | **0.2044** | **0.2366** | **0.0500** | **0.0244** | 23.6447 | **0.6098** |
| T5 BS | 0.2144 | 0.2007 | 0.0409 | **0.0207** | 21.5518 | 0.5827 |
| T5 $\text{Top}_{pk}$ | **0.2236** | **0.2454** | **0.0466** | **0.0228** | 7.2996 | 0.6715 |
| T5 $\text{Top}_k$ | 0.2076 | 0.2384 | 0.0376 | 0.0136 | **5.3002** | **0.6922** |
| T5 $\text{Top}_p$ | **0.2159** | **0.2390** | **0.0430** | 0.0184 | **6.8353** | **0.6743** |
| DialoGPT BS | **0.2192** | 0.2272 | **0.0528** | **0.0312** | 21.6800 | 0.5280 |
| DialoGPT $\text{Top}_{pk}$ | **0.2132** | **0.2444** | **0.0437** | **0.0201** | 6.4158 | 0.6737 |
| DialoGPT $\text{Top}_k$ | 0.2023 | 0.2302 | 0.0320 | 0.0134 | **4.7278** | **0.6956** |
| DialoGPT $\text{Top}_p$ | 0.2093 | **0.2397** | 0.0385 | 0.0159 | **6.1472** | **0.6740** |
| GPT-2 BS | **0.2195** | 0.2132 | **0.0516** | **0.0313** | 23.0605 | 0.5402 |
| GPT-2 $\text{Top}_{pk}$ | **0.2055** | **0.2342** | 0.0384 | 0.0173 | 6.5899 | 0.6832 |
| GPT-2 $\text{Top}_k$ | 0.1956 | 0.2271 | 0.0345 | 0.0153 | **4.7624** | **0.7022** |
| GPT-2 $\text{Top}_p$ | 0.2014 | **0.2329** | **0.0388** | **0.0177** | **6.1944** | **0.6846** |

Table 6: The results computed on the $\text{Best}_{\text{M+D}}$ generations (2500 CN for each model-decoding mechanism combination).

**BERT.** With BS, BERT achieves the best or second best result on all human evaluation metrics, except for specificity. For BERT the best decoding is $\text{Top}_p$: it is the best performing on overlap metrics and the second best for novelty. It achieves good results both on syntactic metrics and human evaluation too.

**T5.** For T5, $\text{Top}_{pk}$ is the best decoding mechanism. It records the best results for overlap metrics and toxicity, and it has good results on syntactic and human evaluation metrics. For what regards $\text{Top}_k$, it is the best for diversity, while $\text{Top}_p$ is good on the syntactic metrics. BS achieves good results on human evaluation, except for specificity and is-best.

**GPT-2.** With $\text{Top}_{pk}$, GPT-2 performs well on ROUGE, BLEU-1, suitableness, grammaticality, and choose-or-not. With $\text{Top}_p$, GPT-2 records the second best result on BLEU scores and diversity metrics. With BS the model has the best performance on overlap metrics (except BLEU-1), and on suitableness, grammaticality, and choose-or-not, but it has also the worst results on diversity metrics. Above all, $\text{Top}_k$ is the decoding achieving the best compromise, reaching the best results for the diversity metrics, and with a superior specificity score (3.15) that is corroborated by the good performance on the other human evaluation metrics.

**DialoGPT.** $\text{Top}_k$ performs best with diversity metrics and specificity; it records the second high-

| | Toxicity | Syntactic metrics | | | |
|---|---|---|---|---|---|
| | - | ASD | MSD | NST | n |
| BART BS | 0.4870 | 3.8919 | 4.6757 | **1.8919** | 37 |
| BART Top$_{pk}$ | **0.3911** | 4.3592 | 4.9483 | 1.6207 | 58 |
| BART Top$_{k}$ | **0.4021** | **4.3798** | **5.0656** | 1.7377 | 61 |
| BART Top$_{p}$ | 0.4263 | **4.5038** | **5.0909** | **1.7727** | 44 |
| BERT BS | **0.3954** | 4.5556 | 5.3750 | 1.9167 | 24 |
| BERT Top$_{pk}$ | **0.4026** | **5.2299** | 6.2069 | 2.1379 | 58 |
| BERT Top$_{k}$ | 0.4157 | 4.8969 | **6.2969** | **2.5625** | 64 |
| BERT Top$_{p}$ | 0.4032 | **5.1019** | **6.2963** | **2.2593** | 54 |
| T5 BS | 0.4127 | 4.4844 | 4.6562 | 1.3438 | 32 |
| T5 Top$_{pk}$ | **0.3211** | **4.7754** | 5.3768 | **1.7826** | 69 |
| T5 Top$_{k}$ | **0.3441** | 4.6767 | **5.4200** | 1.7400 | 50 |
| T5 Top$_{p}$ | 0.3934 | **4.7245** | **5.5918** | **1.8367** | 49 |
| DialoGPT BS | 0.3635 | 4.2340 | 5.1277 | 1.8723 | 47 |
| DialoGPT Top$_{pk}$ | **0.3361** | 4.7264 | 5.5094 | 1.7547 | 53 |
| DialoGPT Top$_{k}$ | 0.3482 | **4.9333** | **6.1778** | **2.0000** | 45 |
| DialoGPT Top$_{p}$ | **0.3274** | **4.7970** | **5.5273** | **1.9636** | 55 |
| GPT-2 BS | 0.3540 | **4.8901** | 5.3617 | 1.4468 | 47 |
| GPT-2 Top$_{pk}$ | **0.3119** | 4.2530 | 5.4182 | 2.4000 | 55 |
| GPT-2 Top$_{k}$ | **0.3416** | **4.6771** | **5.8627** | **2.5686** | 51 |
| GPT-2 Top$_{p}$ | 0.3659 | 4.5663 | **5.7447** | **2.4894** | 47 |

Table 7: The results of the toxicity and the syntactic metrics calculated on the subset employed for the human evaluation and grouped by each combination of model and decoding mechanism. The size of each group is showed in the column "n".

est score on grammaticality. Top$_p$ has the second best result on diversity metrics and BLEU scores. BS is the best on overlap metrics (except BLEU-1), and also on almost all human evaluation metrics: it is the worst on specificity and on diversity metrics. Top$_{pk}$ is the one working best with DialoGPT, since it reaches very good scores with human and overlap metrics, and this does not invalidate diversity, for which it ranks 3rd out of 4.

### A.3 Examples of the generations

> **Hate speech**: Islam is a violent religion that promotes terror and war.
>
> **DialoGPT - BS**: *The vast majority of Muslims in the world are peaceful and friendly people. I don't see how you can say such a thing about them.*
>
> **BART - Top$_k$**: *The vast majority of Muslims are peaceful people who have lived in peace and harmony for many years. How can you say this about a religion that promotes terror and war?*
>
> **BERT - Top$_k$**: *islam is a religion of peace, and it is not a peaceful religion of violence. it is about peace and understanding of the world's problems, and peace is about bringing peace and peace.*

| | Human evaluation | | | | | |
|---|---|---|---|---|---|---|
| | SUI | SPE | GRM | CHO | BEST | n |
| BART BS | 3.7568 | 2.5270 | 4.9459 | 0.8108 | 0.2297 | 37 |
| BART Top$_{pk}$ | **3.7931** | **2.6121** | **4.9483** | **0.8534** | **0.3707** | 58 |
| BART Top$_{k}$ | **3.9672** | **2.5410** | 4.9016 | **0.8607** | **0.2951** | 61 |
| BART Top$_{p}$ | 3.5682 | 2.5114 | **4.9659** | 0.8182 | 0.1477 | 44 |
| BERT BS | **3.5208** | 2.5208 | **4.7917** | **0.7708** | **0.1250** | 24 |
| BERT Top$_{pk}$ | **3.1810** | 2.5776 | **4.2328** | 0.7155 | 0.1121 | 58 |
| BERT Top$_{k}$ | 3.0312 | **2.7031** | 4.1562 | 0.6797 | 0.1016 | 64 |
| BERT Top$_{p}$ | 3.0370 | **2.7130** | 4.1296 | **0.7407** | **0.1574** | 54 |
| T5 BS | **3.5781** | 2.2812 | **4.8438** | **0.7656** | 0.0781 | 32 |
| T5 Top$_{pk}$ | **2.8841** | **2.4928** | 4.5870 | **0.6667** | **0.1014** | 69 |
| T5 Top$_{k}$ | 2.4600 | 2.3200 | 4.6400 | 0.5600 | 0.0500 | 50 |
| T5 Top$_{p}$ | 2.8163 | **2.4388** | **4.7449** | 0.6122 | **0.1224** | 49 |
| DialoGPT BS | **4.1596** | 2.6064 | **4.9894** | **0.8511** | **0.3085** | 47 |
| DialoGPT Top$_{pk}$ | **3.3679** | **2.8019** | 4.8396 | **0.7830** | **0.2736** | 53 |
| DialoGPT Top$_{k}$ | 3.1333 | **2.9222** | **4.8556** | 0.7333 | 0.2111 | 45 |
| DialoGPT Top$_{p}$ | 2.9727 | 2.7000 | 4.8455 | 0.7091 | 0.1909 | 55 |
| GPT-2 BS | **4.3085** | 2.5000 | **4.9681** | **0.8830** | **0.2766** | 47 |
| GPT-2 Top$_{pk}$ | **3.4909** | 2.8000 | **4.8727** | **0.8273** | **0.2273** | 55 |
| GPT-2 Top$_{k}$ | 3.0392 | **3.1471** | 4.8431 | 0.7255 | 0.2549 | 51 |
| GPT-2 Top$_{p}$ | 3.4362 | **3.0638** | 4.7872 | 0.7447 | 0.3298 | 47 |

Table 8: For each model-decoding mechanism combination, these are the results of the metrics for the human evaluations. The size of each combination is showed in the column "n".

> **T5 - Top$_p$**: *The vast majority of Muslims in the UK are peaceful people who share many values and respect.*
>
> **GPT-2 - Top$_p$**: *This is not true, the vast majority of Muslims is peaceful and peaceful Muslims are happy to live in any country they wish. They do not need to be a part of a hostile state and they should be treated as equals in society.*

In these examples we can observe that the CN obtained with GPT-2, albeit not perfect, is the most complete and accurate. Those generated with T5 and DialoGPT are suitable as well. On the other hand, the CNs obtained with BART and BERT present some problems: the former has a clear contradiction in it and the latter is illustrative of some of the problems detected in BERT's generations such that it is syntactically complex but very vague and repetitive, and it ends up not really conveying a clear message.

> **Hate speech**: The migrant crisis is not the fault of Western Europeans. Migrants are to blame for immigration.
>
> **DialoGPT - Top$_k$**: *What if an entire religion was to blame only for the migration crisis? So you think a large number of Muslims are responsible for all the problems of our world?*

**BART - Top$_p$**: *The migrant crisis is caused by a lack of integration and lack of knowledge of the world.*

**BERT - BS**: *many migrants come here to work and provide for their families. they are our brothers and sisters, and we should welcome them.*

**T5 - Top$_k$**: *What evidence do you have for the evidence that migrants are to blame? Not all migrants have done anything wrong.*

**GPT-2 - Top$_k$**: *The EU has put in place laws to combat illegal immigration and to prevent crimes committed by migrants. They have a duty to help stop crime and do what is necessary to solve the problems within their borders. Migrants are very intelligent and do what they can to support their families while they study and work.*

Once again, GPT-2 and DialoGPT generations where chosen as the best CNs by the annotators. Moreover, these CNs are all generated via Top$_k$ decoding mechanism. BART and T5 generated CNs are acceptable while requiring some post-editing to be employed. Finally, BERT's output is still very vague: we can notice the presence of the recurring expression *our brothers and sisters*.

### A.4 Additional material for LOTO experiments

Table 9 displays the distribution of the examples with respect to the targets, in the reference dataset and in the configurations for the LOTO experiments (Section 5.3).

Table 10 presents the detailed results for the novelty of the reference CNs discussed in Section 5.3, while the RR for the CNs generated with the LOTO models and for the reference CNs are shown in Table 11. The rankings for these two RR computations are the same, and the ranges are almost overlapping. This means that leaving one target out does not impact the intra-corpora repetitiveness: instead, the CNs generated with a LOTO model gain a lower RR than the reference CNs. For the target MUSLIMS a high RR is recorded, both in candidate and in the reference CNs. A high repetitiveness in the data for this target can contribute to the good results observed on overlap metrics too (Table 3 in Section 5.3): it is easier that two outputs are similar if they use a limited and repeated number of words.

| Target | Samples in original dataset | Samples in LOTO experiment |
|---|---|---|
| JEWS | 594 | 600 |
| LGBT+ | 617 | 600 |
| MIGRANTS | 957 | 600 |
| MUSLIMS | 1335 | 600 |
| WOMEN | 662 | 600 |
| DISABLED | 220 | 220 |
| POC | 352 | 352 |
| other | 266 | 157 |
| Total | 5003 | 3729 |

Table 9: The targets coverage in the reference dataset (Fanton et al., 2021) and in the LOTO configurations.

| generation / training | JEWS | LGBT+ | MIGRANTS | MUSLIMS | WOMEN |
|---|---|---|---|---|---|
| JEWS | - | 0.775 | 0.780 | **0.761** | 0.780 |
| LGBT+ | 0.781 | - | 0.783 | 0.765 | **0.763** |
| MIGRANTS | 0.782 | 0.775 | - | 0.764 | 0.777 |
| MUSLIMS | 0.775 | 0.770 | 0.769 | - | 0.776 |
| WOMEN | 0.789 | 0.771 | 0.783 | 0.775 | - |

Table 10: The novelty of the reference CNs in the data from Fanton et al. (2021) (*generation*) with respect to the training data for the LOTO models (*training*).

| Target | RR *reference CN* | RR *candidate CN* |
|---|---|---|
| JEWS | 5.071 | 4.796 |
| LGBT+ | 4.489 | 4.620 |
| MIGRANTS | 4.381 | 4.707 |
| MUSLIMS | 5.244 | 5.314 |
| WOMEN | 4.547 | 4.632 |

Table 11: The RR computed on the reference CN (pertaining the test set) and on the CN generated with the LOTO models.

### A.5 APE Experiment Details

The dataset by (Fanton et al., 2021) contains three versions of the same CN: the original CN generated by a GPT-2 model (CN$_{or}$), the expert post-edited versions obtained during the human-in-the-loop cycles (CN$_{pe*}$), and the final version rechecked by NGO experts (CN$_{pe}$).

For fine-tuning our APE model, we have thus used the triplets $<HS, CN_{or}, CN_{pe}>$ and $<HS, CN_{pe*}, CN_{pe}>$. In this way, we managed to roughly double the number of the post-edit training samples, which is highly beneficial for a better model. When we filtered the triplets with a positive

TER score between $CN_{ed}$ and $CN_{pe}$, or $CN_{or}$ and $CN_{pe}$, we obtained 4185 training, 596 test, and 568 validation samples following the partition used in the first set of experiments as described in Section 3.1. Finally, the best fine-tuning configuration of the GPT-2 medium model for APE was obtained with a learning rate of 2e-5 for 3 epochs resulting in 3.34 train loss and 1.23 eval loss.