

# IDT: Dual-Task Adversarial Rewriting for Attribute Anonymization

Pedro Faustini<sup>1</sup>, Shakila Mahjabin Tonni<sup>2</sup>, Annabelle McIver<sup>1</sup>,  
Qiongkai Xu<sup>1</sup>, and Mark Dras<sup>1</sup>

<sup>1</sup>School of Computing, Macquarie University  
pedro.faustini@mq.edu.au, annabelle.mciver@mq.edu.au,  
qiongkai.xu@mq.edu.au, mark.dras@mq.edu.au

<sup>2</sup>Data61, CSIRO  
shakila.tonni@data61.csiro.au

*Natural language processing (NLP) models may leak private information in different ways, including membership inference, reconstruction, or attribute inference attacks. Sensitive information may not be explicit in the text, but hidden in underlying writing characteristics. Methods to protect privacy can involve using representations inside models that are demonstrated not to detect sensitive attributes or—for instance, in cases where users might be at risk from an untrustworthy model, the sort of scenario of interest here—changing the raw text before models can have access to it. The goal is to rewrite text to prevent someone from inferring a sensitive attribute (e.g., the gender of the author, or their location by the writing style) while keeping the text useful for its original intention (e.g., the sentiment of a product review). The few works tackling this have focused on generative techniques. However, these often create extensively different texts from the original ones or face problems such as mode collapse. This article explores a novel adaptation of adversarial attack techniques to manipulate a text to deceive a classifier w.r.t. one task (**privacy**) while keeping the predictions of another classifier trained for another task (**utility**) unchanged. We propose IDT, a method that analyses predictions made by auxiliary and interpretable models to identify which tokens are important to change for the privacy task, and which ones should be kept for the utility task. We evaluate different datasets for NLP suitable for different tasks. Automatic and human evaluations show that IDT retains the utility of text, while also outperforming existing methods when deceiving a classifier w.r.t. a privacy task.*

## 1. Introduction

There are many different ways that NLP models can be vulnerable to leaking private information, and many ways that developers or users of NLP systems can attempt to preserve privacy (Sousa and Kern 2023). Much of the focus has been on protecting the training set to be resistant to, for example, membership inference or reconstruction;

---

Action Editor: Xuanjing Huang. Submission received: 24 June 2024; revised version received: 29 May 2025; accepted for publication: 6 June 2025.

<https://doi.org/10.1162/COLLa.17>

such work has often used a framework with mathematical guarantees like central Differential Privacy (DP), for instance via training techniques like DP-SGD (Abadi et al. 2016). Privacy violations via *inference* are another class of attack: While the ability to infer potentially sensitive information (say, age or gender) has long been known (Schler et al. 2006, for example), there has been recent concern at the demonstrated ease of this kind of inference via large language models (Staab et al. 2024a). The sort of scenario of interest here might include an individual who posts a review or comments on a Web site or social media forum like Reddit, but does not wish a sensitive attribute that may enable identification to be discoverable; the same techniques that facilitate automatically determining, for example, aggregate sentiment, can also automatically find traces of these sensitive attributes.

Whereas there is a broad class of privacy protections against this kind of inference that involve making a model trustworthy, and learning latent representations that although still useful for the primary or utility task (say, sentiment classification) do not leak sensitive information (Li, Baldwin, and Cohn 2018; Yu et al. 2022, for example), we are interested in the kind of scenario where individuals may be at risk from a model that cannot be trusted in this way.

Another kind of protection, the one of interest in this article, is in modifying *raw text* before a model gains access to it. In NLP, works have largely focused on generative approaches that aim to provide (local) DP guarantees of a general sort (Krishna, Gupta, and Dupuy 2021; Weggenmann et al. 2022; Chen et al. 2023). However, these often result in texts that differ extensively from the original and typically do not have a specific sensitive attribute they aim to protect.

A small number of other studies have aimed to change the text in a way that does protect some specific attribute. Xu et al. (2019) proposed an adversarial trainer inspired in Generative Adversarial Networks (Goodfellow et al. 2014) in which a linear classifier sends a signal to a Transformer (Vaswani et al. 2017) to paraphrase a text towards a different class. Another approach, from Tokpo and Calders (2022), trains a model to replace tokens with similar ones, but it needs to adopt a soft sampling to allow gradients to backpropagate. Both approaches, although potentially producing natural-looking text that achieves the goal, face challenges typical of generative models such as mode collapse, which is the case when mostly non-diverse samples are generated (Huijben et al. 2023).

In this article, then, we have a similar goal: a method that only minimally rewrites a text from an author's intended original, maintaining utility while empirically preserving attribute privacy in the sense of Coavoux, Narayan, and Cohen (2018) or Li, Baldwin, and Cohn (2018). Our aim is to define a method that is robust and effective across datasets and tasks and that does not suffer from issues like mode collapse. To do this, we draw on the notion of adversarial attacks in a novel way. Adversarial attacks fool classification models by making subtle changes to their inputs. While they have been widely studied in the context of classification tasks, they have not been applied in scenarios we refer to as **dual-task**, where a piece of text is associated with labels for two distinct classification tasks. In the kinds of scenario of interest here, we apply them to fool one kind of classifier (for a sensitive attribute) while retaining utility on another classifier.

We thus propose **Interpretable Dual-Task—IDT**. In addition to adversarial attacks, we take inspiration from membership inference attacks (Shokri et al. 2017) and train *auxiliary models*, which mimic the target models. We analyze the predictions made by them with an interpretable model that ranks words according to their importance for a given classification task. This way, we can find important words for the privacy and the

utility tasks. We then modify relevant tokens for the private task, but keep the important ones for the utility task unchanged.

Our contributions are as follows:

- We design a method based on adversarial attacks for rewriting text, IDT, such that the rewritten text preserves performance on some utility task, as well as other measures of semantic similarity, while avoiding detection of some potentially sensitive attribute.
- We carry out an extensive evaluation of IDT with other types of text rewriting for privacy purposes. We show that IDT in general outperforms these other approaches, often by a large margin; moreover, we also show that some classes of empirical privacy protection methods do not protect in our kind of scenario.

## 2. Related Work

In the following, we briefly review some relevant NLP work on privacy, followed by work on adversarial attacks, on which our method is based. These subsequent studies lie on single classification problems, and often in binary classification tasks. Therefore, we identify a gap in the literature concerning the privacy-utility trade-off in multi-class problems, as well as cases where instances may belong to several classes.

### 2.1 Privacy in NLP

As noted in Section 1, our task of interest concerns rewriting texts in order to conceal an attribute of the author, on which there has been relatively little work. We discuss this first, but also position it with respect to previous works with related goals, in particular those that aim to ensure privacy by modifying the embedding space that represents the texts, or by altering the text with some obfuscation technique such as differential privacy in order to provide some more general notion of privacy. (Sousa and Kern [2023]) give a detailed survey of these related approaches; we note in this section just some particular instances.)

*Rewriting Texts for Author Attribute Concealment.* The works that have the same goal as ours—that is, to rewrite raw text input with the goal of concealing a sensitive attribute from detection—come from Xu et al. (2019) and Tokpo and Calders (2022). The first explored how Back-translation reduces the leakage of sensitive information. Their approach is heavily inspired by generative adversarial networks and thus sustaining training stability to avoid mode collapse is difficult. In terms of evaluation, it was assessed on three datasets designed for only binary classification problems, and the utility task was restricted to sentiment analysis; we are interested in evaluating contexts beyond this. A similar issue with potential mode collapse happens with the work in progress presented in Tokpo and Calders (2022). It changes the style of a text w.r.t. a task by replacing individual tokens. However, their method requires a soft sampling, and finding the appropriate  $\tau$  temperature parameter is challenging. In contrast to these, we use an optimization-based adversarial attack approach, that can be expected to produce rewritten texts in a more reliable fashion.

Staab et al. (2024a) also noted that ChatGPT can infer underlying personal attributes that are not explicit in the text, such as the gender of the writer. Their application domain for this was user profiles consisting of multiple texts from a Reddit dataset, so the data available for inference differs somewhat from the usual. Building on this (and contemporaneous with our work), Staab et al. (2024b) also established that LLMs can be used to anonymize text in this same context of user profiles. Even though LLMs can be used without further fine-tuning, prompt engineering is an issue since the prompt may need to be heavily adjusted according to the task, and the authors need to query LLMs in several iterations.

We remark that all of these are frameworks that use empirical evaluations to assess the level of privacy achieved, as is ours.

*Rewriting Texts Using Differential Privacy.* Works within a local DP framework create alternative texts by adding noise to an encoder or decoder (Igamberdiev, Arnold, and Habernal 2022; Yue et al. 2021), and privacy is provided by guarantees that come with the DP framework, usually characterized via the core  $\epsilon$  and  $\delta$  parameters of the framework. This contrasts with the empirical evaluations above, and the privacy guarantee is not linked to any specific attribute of the text or related (labeled) task.

As some examples, CusText (Chen et al. 2023) and DP-Prompt (Utpala, Hooker, and Chen 2023) replace tokens according to an obfuscated list of semantic similarities, while the approach of Weggenmann et al. (2022) perturbs the latent vectors and later decodes them into text. The texts produced by these approaches often differ greatly from the original (e.g., METEOR similarity for Weggenmann et al. [2022] was very low, 5% or 9% according to the dataset, indicating that many tokens were replaced). Further, it is a more general notion of protection provided here, with Igamberdiev and Habernal (2023) emphasizing the need to be clear on what exactly is being privatized. For instance, paraphrasing has been studied by Ponomareva, Bastings, and Vassilvitskii (2022) by pre-training a T5 model (Raffel et al. 2020) and its tokenizer with DP to prevent training data from leaking when the model generates text. In another example, CusText evaluated privacy under an unlabeled task of predicting the original tokens using a BERT-MLM model against the sanitized sentence. In their survey, Klymenko, Meisenbacher, and Matthes (2022) discuss that practical benefits of DP in NLP applications lie in the individual space: An attacker could still infer tokens from text, but there would be some uncertainty as to whether such token was indeed the original one. However, it is unclear how to control the effect of such sanitization for a specific task of concern in our sort of attribute inference scenario under such a framework.

Also, it is common for the  $\epsilon$  parameter from DP to be large, in the order of hundreds or even thousands, which means low privacy guarantees. Igamberdiev and Habernal (2023) acknowledge that the lowest useful  $\epsilon$  is too high for real-world applications, given that different applications in the literature range  $\epsilon$  from 0.01 to 10. Sousa and Kern (2023) remark that privacy in NLP is exchanged for performance: For example, adding noise to the embeddings may cause semantic disruption, compromising the final results of downstream tasks. To avoid that, in DP-SGD works, such as from Kerrigan, Slack, and Tuyls (2020), non-private datasets are still necessary for pretraining models, and the private data is used in later stages, for fine-tuning.

*Modifying Embedding Representations for Author Attribute Concealment.* Fernandes, Dras, Fernandes, Dras, and McIver (2019), Plant, Gkatzia, and Giuffrida (2021), and Meehan, Mrini, and Chaudhuri (2022), for instance, apply different types of differential privacy noise to various embedding representations in order to prevent sensitive information

from being inferred from the data. Aiming to protect privacy via an empirical demonstration rather than using the mathematical guarantees of differential privacy, on the other hand, Li, Baldwin, and Cohn (2018) trained a generative model in an adversarial fashion in which the model generates a (latent) representation  $\mathbf{h}$  of a text for some utility task (e.g., POS tagging). At the same time,  $\mathbf{h}$  is designed to be a bad representation for sensitive attributes (such as age or gender). Coavoux, Narayan, and Cohen (2018) provide another approach to the same task of producing representations that aim to avoid encoding sensitive attributes. Work has continued in this space, for example, looking at collaborative learning contexts (Chen, Qiang, and Ruan 2022) and at applying noise to clustered representations (Zhou et al. 2023). Apart from DP guarantees that come with the work of Fernandes, Dras, and McIver (2019), Plant, Gkatzia, and Giuffrida (2021), and Meehan, Mrini, and Chaudhuri (2022), the evaluation here is again empirical.

These kinds of approaches are suitable for scenarios where, for example, a system should use representations that fulfill its primary functionality well but where the system creator wants to convince users that the system’s decisions will not take account of their sensitive attributes. This contrasts with our perspective in this article, from the point of view of a user who may not trust a system creator.

*Text Sanitization and Text Anonymization.* One task related to the above is text sanitization, in which a model rewrites the input by removing or replacing personally identifiable information (PII) (Albanese, Ciolek, and D’Ippolito 2023). The difference is that the property to keep private is not a task, but individual tokens. As a consequence, the narrow vocabulary that can be changed, coupled with the limited options for the replacements (usually a predefined token), may not be enough to obfuscate high-level properties one may wish to hide. Closely linked is text anonymization (Lison et al. 2021), the task of editing a text document to prevent the disclosure of personal information. The work in this space typically has a broader focus than text sanitization, sometimes working with formal notions of disclosure risk, but like text sanitization often has a focus on identifying specific spans of information like PII, as exemplified by the Text Anonymization Benchmark (Pilán et al. 2022).

## 2.2 Adversarial Attacks

Adversarial attacks are methods that strategically modify input text to fool a model into making an incorrect prediction. There are now a number of adversarial attacks against text in the literature: In terms of overviews, Zhang et al. (2020b) and Qiu et al. (2022) have provided surveys, while Dyrmishi, Ghamizi, and Cordy (2023) empirically compared some of the major methods. Morris et al. (2020b), in designing a common framework for adversarial attacks called TextAttack,<sup>1</sup> showed how previously disparate adversarial attack methods could be thought of in a unified way, consisting of four types of components: A goal function, operationally specifying the method goal such as changing classifier prediction or changing words used in translation; a set of constraints reflecting desiderata for adversarial text quality, such as semantic similarity to the original text or grammaticality; transformations, the set of allowable perturbation types; and search methods, the approach (e.g., genetic algorithm) to finding an adversarial text using the transformations that satisfies the constraints and the goal function. Adversarial attacks are shown transferable from shadow models to the original models (He et al.

---

<sup>1</sup> <https://github.com/QData/TextAttack>.

2021). We discuss below the three specific adversarial attacks that we use as the basis for our privacy-protecting method; all three are implemented in the TextAttack framework.

TextBugger (Li et al. 2019) works by comparing the prediction before and after removing a word or character to measure its influence. The utility is measured in terms of four similarity metrics between the sentences, which should be above a specific threshold. Moreover, the authors carried out a human evaluation which concluded that the sentences retained their sentiment/toxicity.

Another popular attack is TextFooler, by Jin et al. (2020). It replaces words according to a list of constraints as cosine similarity and maintains POS tags.<sup>2</sup> However, their evaluation compared the accuracy scores of original and adversarial sentences for one task per dataset. Utility was measured by reporting the ratio of perturbed words and the semantic similarity between the texts.

Garg and Ramakrishnan (2020) proposed BAE, which masks tokens and uses the BERT-MLM to create alternatives to replace them. Utility, as in the previous studies, is assessed by calculating semantic similarity, with the addition of humans evaluating how suspicious the sentences were altered by a machine and how they rate them w.r.t. their sentiment—which is the same task used for the attacks.

### 3. Problem Framing

*Threat Model.* Our threat model is broadly the free text inference threat outlined in Staab et al. (2024a), although their focus is prompt-based LLMs.

The adversary wants to infer, from text produced by a user (this may often be publicly available text such as comments on a social media site, but possibly also (il)legally obtained records of internal communications) sensitive attributes of that user, such as age or location.

This adversary may be expected to have access to machine learning models, specifically language models, that can make these inferences.

*Problem Definition.* We observed in Section 2 that many techniques for rewriting for privacy can involve radically different text being generated. Our idea is to constrain this by considering a specific utility class label that should be preserved while aiming to hide the private attribute.

We thus define our problem as a dual-task optimization problem: a problem involving two tasks, where these tasks are related to privacy ( $p$ ) and utility ( $u$ ). Let  $D = \{\mathcal{X}, \mathcal{Y}^p, \mathcal{Y}^u\}$  be a dataset where  $\mathcal{X}$  is a collection of texts, and  $\mathcal{Y}^p$  and  $\mathcal{Y}^u$  are their labels for task  $p$  (privacy) and task  $u$  (utility).

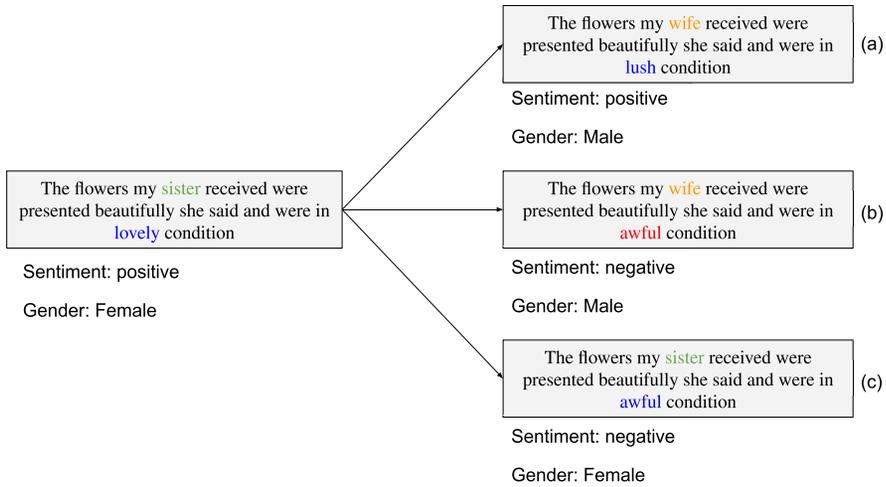
An adversary aims to discover the property (label)  $y^p \in \mathcal{Y}^p$  corresponding to a text  $x \in \mathcal{X}$ . We assume that there is some classifier  $f_p$  that assigns instances  $x \in \mathcal{X}$  to a label  $y_p \in \mathcal{Y}^p$ . At the same time, we assume that the text will be used for some primary utility purpose. Similarly, there is a classifier  $f_u$  that assigns a label  $y_u \in \mathcal{Y}^u$  to instances  $x \in \mathcal{X}$ . A **rewriter** aims at generating a perturbed text  $x'$  from a genuine text  $x \in \mathcal{X}$ .

A *successful* rewriting is defined as  $f_p(x) \neq f_p(x')$  and  $f_u(x) = f_u(x')$ : The utility of  $u$  for  $x'$  is kept while the privacy of  $p$  is retained by forcing  $f_p$  to misclassify  $x'$ .

Intuitively, our problem deals with making minimal editing to texts so that their usefulness remains not merely in terms of how close the attacked sentences are to

---

<sup>2</sup> The TextAttack implementation relaxes this constraint by allowing nouns to be swapped by verbs and vice-versa.



**Figure 1**  
By identifying which words are important to each classification task, one can rewrite the sentence aiming to deceive a classifier w.r.t. a particular task only.

their original counterparts, but for a utility task. For instance, posts in marketplaces are *useful* if they keep their sentiment towards the products, whereas potentially sensitive identifiable information w.r.t. the writer (e.g., their gender, or age) can be hidden. We are principally interested in scenarios where there are machine learning models that can detect the sensitive attribute, and the user might want to avoid such detection. As an extension of the constrained optimization approach to constructing adversarial examples, the aim here is likewise to minimally change the original text.

We show in Figure 1 an example<sup>3</sup> depicting how a single sentence can be associated with different tasks, and how changing particular words may affect the prediction for each task. Without looking at any particular class, a sentence may be rewritten in a way to lose all its semantic content (as in rewriting (b)). This can be avoided by defining one task to change the prediction of another to keep it.

Note that while the novel aspects of our work primarily concern the adaptation of adversarial attack methods to achieve this goal, our problem formulation above does differ in some respects from prior work. This is because our notions of utility and privacy are defined with respect to some concrete (classification) tasks, with prior work differences in focus and framing inferrable from what they evaluate. The closest comparison works, Xu et al. (2019) and Tokpo and Calders (2022), do measure privacy with respect to classification performance on a sensitive attribute. However, Xu et al.’s (2019) primary non-privacy evaluation is on “linguistic quality,” including both automatic metrics (e.g., BLEU, GLEU, and Word Mover Distance) and human evaluation of fluency and relevance, or semantic closeness to original (although a secondary analysis does also look at classification accuracy of a utility task). Tokpo and Calders (2022)

<sup>3</sup> This example comes from a real instance in one of the datasets we use, TrustPilot (Hovy, Johannsen, and Søgaard 2015). In this dataset, there are gender labels (only binary male/female) and sentiment labels. The gender labels predicted here for the rewritten versions comes from a machine learning classifier that makes the most likely inference. A real-world instance of the problems of detecting gender was where it implicitly took place during recruitment processes using machine learning: <https://bit.ly/2ycdnVV>. We posit that a person who wanted to conceal their gender might be willing to consider such a rewriting.

likewise look at semantic closeness to original as measured by cosine similarity. Staab et al. (2024b) likewise consider semantic similarity, but derived from an LLM scorer on a scale of 1–10. Other types of private rewriting within a DP framework, such as CusText (Chen et al. 2023), are not concerned with a specific sensitive attribute, as they typically evaluate the privacy protection only in terms of privacy budget  $\epsilon$ .

Our notion of utility preservation is a coarse-grained one, designed to act as a guard-rail to make sure the rewriting is not too radical; evaluating whether it is successful has the advantage of being straightforward and having a clear interpretation (performance on utility task). Measures like semantic similarity can be less interpretable and linking to human judgments regarding preservation of meaning can be challenging (Morris et al. 2020a),<sup>4</sup> but they do provide a more fine-grained assessment of utility. We use both in this article.

#### 4. IDT Rewriting

IDT<sup>5</sup> modifies important tokens for the privacy task, while keeping important ones for the utility task. We refer to the classifier for the privacy task that is representative of the adversary as IDT’s **target model**, with the aim being to fool this target model while maintaining performance with respect to utility.

For IDT, we take inspiration from membership inference attacks (Shokri et al. 2017) and train shadow (auxiliary) models, whose goal is to mimic the target models. In membership inference attacks, attackers, trained based on the outputs of shadow models, infer if an object was used for training by looking at the class probabilities outputs. In our case, our shadow models, hereafter called **auxiliary models**, are used to find the important words for each classification task.

Our rewriting method has the following assumptions:

- As with membership inference attacks, we assume that the rewriter has data to train auxiliary models.
- The rewriter uses the auxiliary models to create adversarial sentences against the victim.
- Previous work queries the target several times to create the adversarial sentences; IDT creates them offline with auxiliary models. Thus, it only queries the target once per sentence.

Algorithm 1 depicts its pseudocode. It has three main blocks; comments highlight each one.

Lines 1 to 8 select potential words to change, with the core of the choice of words happening between lines 5 and 7. Explanations<sub>priv, util</sub> are maps from a token to its score according to the interpretable model running through the auxiliary models  $\theta_{priv}$  and  $\theta_{util}$ . In our experiments, we used the Layer Integrated Gradients from Mudrakarta et al. (2018), available in the Captum library (Kokhlikyan et al. 2020). We emphasize that any

---

<sup>4</sup> Morris et al. (2020a), in case studies examining adversarial attacks, show that constraints over automated semantic similarity metrics with thresholds chosen in an ad hoc manner do not guarantee that semantics are genuinely preserved. They note: “Human evaluation is needed to find threshold values such that people generally ‘agree’ that semantics is preserved.” Care is also necessary in the framing of the task of gathering human judgments.

<sup>5</sup> <https://github.com/phfaustini/IDT>.

**Algorithm 1:** IDT rewriting.

---

**Data:** Sentence  $X = \{w_1, w_2, \dots, w_n\}$ , ground truths  $y^{p,u}$ , auxiliary models  $\theta_{p,u}$ , word embeddings  $E$  over the vocabulary  $V$ ,  $k$  nearest words to consider, query budget  $q$

**Result:** List of adversarial  $Xs'$

```

/* Searching for potential words in X to replace */
1  $explanations_{priv} \leftarrow explain(\theta_p, X)$ ;  $explanations_{util} \leftarrow explain(\theta_u, X)$ ;
2  $top\_words_{priv} \leftarrow$  Get the top influential words from  $explanations_{priv}$ ;
3  $top\_words_{util} \leftarrow$  Get the top influential words from  $explanations_{util}$ ;
4  $final\_words \leftarrow \emptyset$ ;
5 for  $w \in top\_words_{priv}$  do
6   if not  $w \in top\_words_{util}$  then
7      $final\_words \leftarrow final\_words \cup \{w\}$ ;

/* Searching for the most similar words for each word in final_words */
8  $nearest\_words \leftarrow \{\}$ ;
9 for  $w \in final\_words$  do
10   $nearest\_words_w \leftarrow k$  most similar words to  $w \in V$ ;

/* Searching for adversarial sentences by sampling i words each time */
11  $Xs' \leftarrow \emptyset$ ;
12 for  $i$  in  $len(final\_words)$  do
13   repeat
14      $words\_to\_change \leftarrow sample(final\_words, i)$ ;
15      $X' \leftarrow X$ ;
16     for  $w$  in  $words\_to\_change$  do
17        $w' \leftarrow random(nearest\_words[w])$ ;
18        $temp \leftarrow X'.replace(w, w')$ ;
19       if  $POSTags(temp) == POSTags(X)$  then
20          $X' \leftarrow temp$ ;
21          $Xs' \leftarrow Xs' \cup X'$ ;
22   until  $q$  times;

```

---

interpretable method that assigns scores to tokens can be adopted.<sup>6</sup> Then, the influential words (those assigned a positive value) for the privacy and utility tasks are selected, respectively. Then, only those important for the privacy task, but that do not appear in the utility list, are kept. The rationale is to modify sensitive words for privacy while preserving utility.

Lines 9 to 11 select the  $k$  most similar words for each word in the sentence. In our experiments, we converted words to latent space with GloVe embeddings (Pennington, Socher, and Manning 2014).

Last, lines 13 to 22 search for valid adversarial texts.  $sample(final\_words, i)$  selects a number  $i$  of words in the original sentence  $X$  to be replaced. Adversarial sentences

<sup>6</sup> Other interpretable methods are also possible. One example is LIME (Ribeiro, Singh, and Guestrin 2016); it does not require access to gradients. However, it is a generic method that perturbs the original input to observe changes in the corresponding outputs, hence being slower.

are created by replacing words with similar ones in increasing amounts. We also set the constraint that adversarial texts must match the POS tags of the original one, as several previous works have shown this constraint to be effective (Jin et al. 2020; Garg and Ramakrishnan 2020; Yoo and Qi 2021).

From the generated adversarial Xs, we select the one that deceives the auxiliary model  $\theta_{priv}$  with the highest confidence score, that is, the output probabilities (calculated from the logits) of the auxiliary model.

*Caveats Regarding Interpretability.* The notion of interpretability comes with many caveats (Woods 2022; Rudin et al. 2022). It can vary in terms of degree of interpretability being considered, which may be a function of use case requirements. It can be unreliable, as interpretability methods generally use techniques that are less powerful than the models they are explaining. In particular, interpretability techniques can deteriorate in terms of reliability and compactness when models use a high number of features, have strong feature interactions and complex feature main effects among others (Molnar 2022). NLP interpretability has its own specific issues (regarding, e.g., the functionally groundedness of input feature interpretations) (Madsen, Reddy, and Chandar 2022). Notwithstanding these and other issues, interpretability methods do give a rough idea of particularly salient aspects of the text for us to modify.

## 5. Experimental Setup

*Inference Models.* Our core models for inference (for both privacy and utility tasks) are built using distilled RoBERTa (Liu et al. 2019), pretrained base models from the Transformers library (Wolf et al. 2020) as the architecture for both victim and auxiliary models. We also design a further analysis with differing architecture by using a distilled GPT2 (Radford et al. 2019) as victim.

Following Staab et al. (2024a), we also look at using LLMs for inference. Specifically, we use **ChatGPT** 4o-mini (OpenAI 2023), llama (Dubey et al. 2024) (version 3.2 with 1B parameters), and Gemma (Rivière et al. 2024) (version 2 with 2B parameters). This is a challenging task, since these models are generative, i.e., trained to output free text instead of numerical class numbers as models fine-tuned for classification. We leave the prompts in Appendix E.

*Attributes.* In terms of sensitive attributes to anonymize, we choose three attributes that are widely used in the literature: age, gender, and location (Li, Baldwin, and Cohn 2018; Coavoux, Narayan, and Cohen 2018; Chen, Qiang, and Ruan 2022; Zhou et al. 2023; Staab et al. 2024a). We also choose two other attributes, native language and style, that could conceivably also be considered sensitive and in some circumstances can help to identify (aspects of) an individual. For utility, we choose the very common topic and sentiment.

*Datasets.* We pick datasets suitable for both privacy and utility tasks, according to the attributes above. For the sensitive (privacy) attribute, the adversarial text must lead to a different classification than the original, and for the utility attribute, the classification should remain the same for original and adversarial.

**TrustPilot** (Hovy, Johannsen, and Søgaaard 2015) is a core dataset for investigating privacy, and it has been adopted by much research studying attribute inference attacks (He et al. 2022; Coavoux, Narayan, and Cohen 2018; Li, Baldwin, and Cohn 2018). It contains reviews alongside attributes such as numerical rating, gender, location, and

**Table 1**

Privacy and utility tasks per dataset with the number of classes for each one.

Dataset	Privacy	# Classes	Utility	# Classes
TrustPilot <sub>L</sub>	Location	5	Rating	3
TrustPilot <sub>G</sub>	Gender	2	Rating	3
TrustPilot <sub>A</sub>	Age	2	Rating	3
TOEFL11	NLI	11	Topic	8
Shakespeare	Style	2	Play	17

year. We use the ratings as the utility task, converting the 1–5 scale into bad, mixed, and good categories. For the sensitive attribute, we use gender, age, and location.

We follow previous studies and treat age and gender as binary values. We split the age class between those born before 1967 and after 1977, leaving a 10-year gap in between. A similar approach was used by Li, Baldwin, and Cohn (2018). For location, we also follow Li, Baldwin, and Cohn (2018) and retain English reviews according to the Langid tool (Lui and Baldwin 2012) and ensure the texts are balanced among the five classes (Denmark, German, France, UK, US).

To evaluate generalizability across other attributes, we also experiment with two other datasets. **TOEFL11** (Blanchard et al. 2013) was collected for the task of native language identification (NLI).<sup>7</sup> Each essay was written in English by learners from 11 other languages. Each document is also labeled by one of the 8 topics the essay is about. We set the native language as the sensitive attribute, the topic for utility, and split the essays into sentences. Thus, we evaluate privacy and utility under *multi-class tasks*.

**Shakespeare** (Xu et al. 2012) contains sentences from 17 plays, labeled according to their writing style (modern or old). We use the style as the attribute to change and the play as the one to preserve the classification. We note that utility retention should be harder on this dataset, given its large number of classes.

We split each dataset into target and auxiliary model data. Specific details about the splits can be found in Table A.1 in the Appendix.

For each dataset, we retrieved the texts that were correctly classified for both utility and privacy tasks. They are then passed through the auxiliary models, and adversarial texts are generated based on their explanations. The adversarials that deceived the auxiliary models in the privacy task, but not for utility, are queried against the victim. Table 1 summarizes the privacy and utility tasks for each dataset, alongside the number of classes per task.

*Baseline Rewriting Models.* To the best of our knowledge, there is no system that tackles our specific problem as we have framed it. Therefore, we adopt the Back-translation model from Xu et al. (2019) since it is the closest proposal to ours, as highlighted in Section 2.2. More details about its implementation can be found in Appendix C. We also considered the system of Tokpo and Calders (2022). However, we could not successfully train this on our datasets and tasks without mode collapse,<sup>8</sup> so we do not present results for it. The system of Staab et al. (2024b) also appears to be promising, but the code was

<sup>7</sup> Not to be confused with Natural Language Inference, which is a different NLP task.

<sup>8</sup> For instance, the sentence “draw the curtains, just like that.” collapses to “draw the sword, barlow barlow barlow”

not available at the time of writing. Instead, we implemented our own LLM rewriters inspired by this; we describe these at the end.

For this principal baseline, we re-implemented an **adversarial back-translator** that follows the proposal of Xu et al. (2019). Since the source code is not available, we made the following changes: The parallel corpora were generated with the pretrained MarianMT models discussed below. The Back-translator is a BART model instead of the original (now outdated) Transformer (Vaswani et al. 2017). The adversary is a BART classification head. Following their experiments, the  $\alpha$  parameter was set to 1.0. More details are in Appendix C.

Related to this, we evaluate how ordinary (non-adversarial) **Back-translation** disrupts the texts, an approach that has been tried and used as a baseline by other groups (Prabhumoye et al. 2018, for example), including our primary baseline of Xu et al. (2019). Our Back-translation baselines are two MarianMT pretrained models released by Tiedemann and Thottingal (2020). They were trained with the Opus corpus (Tiedemann 2009). One model translates sentences from English to French, and the other converts French texts back to English. The authors report BLEU scores ranging from 27.5 to 50.5 across 10 test sets for the first translator,<sup>9</sup> and from 26.2 to 57.5 for the second.<sup>10</sup>

For further baselines, we investigate whether solutions to related problems already solve ours; specifically, existing adversarial attacks and text sanitization techniques. We compare IDT to adversarial algorithms provided by TextAttack (Morris et al. 2020b), under different query budgets. High budgets are akin to a brute force search and can be prohibitively costly. By default, TextAttack queries the target model several times, which in reality may not be feasible due to defense mechanisms used by servers. Thus, we also experiment when TextAttack creates adversarial texts against our auxiliary models and then they are tested against the target auxiliary, akin to our methodology for IDT.

We use **TextBugger** (Li et al. 2019), **TextFooler** (Jin et al. 2020), and **BAE** (Garg and Ramakrishnan 2020) as extra baselines. These algorithms were chosen because they alter texts in different levels of granularity: TextFooler replaces words based on similarity constraints, whereas BAE uses a BERT-MLM model to generate replacements for masked tokens. TextBugger performs character-level replacements so that sentences are perceptibly similar.

While there are several approaches to text sanitization in the literature, most do not come with code. We adopt one that does, **CusText** from Chen et al. (2023), setting  $\epsilon = 10$ , although we note that it was not designed for adversarial attacks as in our work.<sup>11</sup> Outside DP, we also adopt **Presidio**<sup>12</sup> and **ZSTS** (Albanese, Ciolek, and D’Ippolito 2023) sanitizers.

Finally, we adopted **ChatGPT 4o-mini** (OpenAI 2023) as a zero-shot rewriter to construct reformulations as the last baseline. In Appendix F, we discuss Gemma (Rivière et al. 2024) (version 2 with 2B parameters) and Llama (Dubey et al. 2024) (version 3.2 with 1B parameters) text reformulations as rewriters.

---

<sup>9</sup> <https://huggingface.co/Helsinki-NLP/opus-mt-en-fr>.

<sup>10</sup> <https://huggingface.co/Helsinki-NLP/opus-mt-fr-en>.

<sup>11</sup> A reviewer has also helpfully pointed out that an intended use of sanitizing text for downstream applications would require many more synthetic examples to be collected for that purpose, so there is not a one-to-one kind of rewriting relationship. We nevertheless include some results and examples here to illustrate that such methods are not suited to our task. We also looked at varying  $\epsilon$ , but the overall conclusion stays the same.

<sup>12</sup> <https://microsoft.github.io/presidio/>.

*Model Training.* Last, for all datasets and for models that require training, we used the Adam optimizer with a learning rate of  $3e-5$ , alongside a linear schedule with a warmup ratio of 10%. Auxiliary and target models were trained for 10 epochs for TOEFL11 and TrustPilot, and for 30 epochs for Shakespeare. The batch sizes for TOEFL11, Shakespeare, and TrustPilot were 16, 8, and 64, respectively. Each experiment used a single GPU.

## 5.1 Automatic Evaluation

This is our primary evaluation, which considers how well the models being compared can fool a classifier that detects sensitive attributes while producing rewritten text that maintains utility, in terms of both our coarse-grained (utility class) and fine-grained (semantic similarity) senses in Section 3.

*Rewrite.* We evaluate the success of the rewriting in terms of *rewrite success* (**RS**), which, as defined in Section 3, is built based on the outcomes for both tasks. There is a success when the adversary (in the form of the privacy classifier) is deceived for the privacy task, but the prediction for the utility task remains unchanged. This metric arises fairly naturally from the problem definition in Section 3, and is similar to rewrite success metrics in the single-task case, such as that of Jin et al. (2020), which measured the rewrite success as the difference between the original and after-attack accuracies, and that of Garg and Ramakrishnan (2020), where additionally the attack was considered a failure if all tokens were changed. We, on the other hand, need to take into account the rewrite success under the dual-task nature of our problem. We thus also measure the *utility retention* (**UR**), which is the ratio of unchanged predictions between original and adversarial texts for the utility task. Both RS and UR are computed over original samples correctly classified.

As discussed in Section 3, we also look at semantic similarity metrics as fine-grained measures of utility preservation. Specifically, we measure the BERTScore (Zhang et al. 2020a) with original text as reference, and also the cosine similarity over Universal Sentence Encoder (Cer et al. 2018) embeddings. We consider BERTScore as our primary fine-grained metric, as Zhang et al. (2020a) demonstrated that it was particularly robust to adversarial paraphrases.

*Text Modification.* As in previous studies, we analyze the extent to which the adversarial texts have changed w.r.t. the original ones. We measure the ratio of the Levenshtein distance to the length of the sentences, Jaccard, and Meteor metrics, as well as the proportion of tokens matching POS tags between texts. We also assess the grammar of the texts, judged by a RoBERTa-large classifier trained on the CoLA dataset (Warstadt, Singh, and Bowman 2019) by Krishna, Wieting, and Iyyer (2020). These measurements help to evaluate how close original and adversarial samples are to each other.

## 5.2 Human Evaluation

Our secondary evaluation is in the form of two human judgment tasks regarding:

1. how the sentences retain their utility by asking humans to classify the perturbed texts amongst the utility classes; and
2. how humans rate the sentences regarding grammar and fluency.

The first is similar to the human evaluation conducted by Garg and Ramakrishnan (2020), where judges had to choose the correct class label from a given set; and the second one is modeled on that of Xu et al. (2019), where judges rate the generated sentences from 1 (“Not in the form of human language”) to 5 (“Without any grammatical error”). We highlight that, unlike prior works, we have an explicit class to assess utility, and therefore we do not rely on abstract concepts such as “naturalness” (Garg and Ramakrishnan 2020), “meaning or natural sentence” (Jin et al. 2020), or asking humans to rank “semantic similarity” (Ye et al. 2022).

For UR, we randomly sampled 50 original sentences from TrustPilot and TOEFL11 datasets, alongside adversarial counterparts generated by IDT, adversarial Back-translation, and TextFooler, summing to 200 sentences total. We left the Shakespeare dataset out of this study because it is unlikely that a human can effectively classify a single random sentence into a Shakespeare play, even if they have read the plays. For TrustPilot, we simply ask them to rate if the review is positive or negative. Finally, for TOEFL11, they have to discern if the sentence comes from an essay talking about *cars* or *students & learning*, which are two of the possible eight topics for an essay.

We also only sampled sentences that delivered successful rewritings for all algorithms. We didn’t enforce the same restriction for the second task. Instead, we sampled 20 sentences from each one of the three rewritings, coupled with their original counterparts, regardless of their rewrite success ( $40 * 3 = 120$  sentences). We thus conducted the assignment with alongside the previous 200 sentences from task *i* (320 sentences total).

We conducted all tasks on Amazon Mechanical Turk, with instructions as given in Appendix A. Each sentence was rated by 3 annotators, and we paid them US\$0.13 per HIT. Each HIT takes 30 seconds; therefore they were paid around \$15.60/h, or roughly the Australian national minimum wage for the fiscal year 2023–24,<sup>13</sup> set at A\$23.23/h. Annotators were required to be proficient in English.

## 6. Results

### 6.1 Inference Model Performance

For each dataset, we trained four core RoBERTa models, two for privacy (one main and one auxiliary) and the other two for the utility task (again, one main and one auxiliary). Given that the two main and two auxiliary models have their distinct training sets, which reduces the training data, we, therefore, do not aim to achieve state-of-the-art models that were trained with the whole datasets, but models strong enough to perform satisfactorily across the tasks. Table 2 reports their accuracy on the same test set. We observe a very similar classification capacity for all tasks in all datasets.

We also included results for the LLM inference models in Table 2. We observe that the performance of LLMs across all tasks and all datasets is inferior to fine-tuned models. One could consider ChatGPT’s accuracy in the privacy task to be close enough to the fine-tuned models (above 80%) for the Shakespeare dataset, but its accuracy in identifying the play is much poorer. One reason is that classifying English sentences as modern or traditional is simpler than recognizing which play the sentence comes from in a zero-shot fashion.

---

13 <https://www.fwc.gov.au/agreements-awards/minimum-wages-and-conditions/national-minimum-wage>.

**Table 2**

Accuracy of the target and auxiliary models on the test sets. “Both” means samples were correctly classified in both tasks. We highlight in bold the best scores for the target model’s test set.

Dataset	Model	Privacy	Utility	Both
TrustPilot <sub>L</sub>	RoBERTa	<b>.5380</b>	<b>.9350</b>	<b>.5080</b>
	RoBERTa (aux.)	.5260	.9380	.4970
	ChatGPT	.4350	.8810	.3920
	Llama	.2140	.7000	.1500
	Gemma	.3110	.8790	.2730
TrustPilot <sub>G</sub>	RoBERTa	<b>.7298</b>	<b>.9606</b>	<b>.7018</b>
	RoBERTa (aux.)	.7228	.9586	.6940
	ChatGPT	.6742	.9242	.6230
	Llama	.5186	.6974	.3668
	Gemma	.7592	.9220	.6902
TrustPilot <sub>A</sub>	RoBERTa	<b>.7862</b>	<b>.9578</b>	<b>.7532</b>
	RoBERTa (aux.)	.7932	.9576	.7604
	ChatGPT	.6158	.9176	.5646
	Llama	.3036	.6922	.2130
	Gemma	.4754	.9130	.4316
TOEFL11	RoBERTa	<b>.3767</b>	<b>.8405</b>	<b>.3195</b>
	RoBERTa (aux.)	.3549	.8391	.3017
	ChatGPT	.1928	.8201	.1587
	Llama	.1215	.3706	.0458
	Gemma	.0969	.6408	.0613
Shakespeare	RoBERTa	<b>.8520</b>	<b>.3797</b>	<b>.3280</b>
	RoBERTa (aux.)	.8467	.3742	.3167
	ChatGPT	.8155	.3155	.2490
	Llama	.4840	.0808	.0387
	Gemma	.7205	.0745	.0503

Llama and Gemma perform even worse, albeit Gemma outperforms Llama overall. Still, with these results, for the remainder of the article we carry the adversarial attacks against the RoBERTa fine-tuned models, whose performance on classification tasks are superior.

**6.2 Rewrite Success**

Table 3 brings the results in terms of rewrite success (RS), utility retention (UR), and our two measures of semantic similarity, BERTScore (primary) and USE cosine similarity.<sup>14</sup> In summary, IDT presents competitive performance across all the datasets, showing it is a versatile technique. Specifically, IDT achieves the biggest UR and RS for all flavors of the TrustPilot dataset, substantially ahead of all baselines. IDT is particularly effective for TrustPilot<sub>L</sub>, which holds the largest number of classes among the flavors. Across all tasks, it has a high BERTScore, the second-highest mean of 0.9701 (after the 0.9731 of

<sup>14</sup> We also include the privacy accuracy that is a component of RS.

**Table 3**

Rewrite success (**RS**), utility retention (**UR**), accuracy on the privacy task (**PA**), as well as semantic similarity scores measured by USE cosine similarity (**CS**) and BERTScore (**BS**) for each dataset under different attacks. The scale is in the range of 0 to 1.

Dataset	Rewrite Method	RS $\uparrow$	UR $\uparrow$	PA $\downarrow$	CS $\uparrow$	BS $\uparrow$
TrustPilot <sub>L</sub>	IDT	<b>.6166</b>	<b>1.0000</b>	.3833	.2341	.9756
	Adv. Back-translation	.1003	.1338	<b>.2500</b>	.0219	.7133
	Back-translation	.2594	.9782	.7366	.2276	.9600
	TextFooler	.4427	.9872	.5521	.2289	.9757
	TextBugger	.5375	.9875	.4600	.2372	.9742
	BAE	.4070	.9867	.5929	.2319	<b>.9804</b>
	Presidio	.2236	.9868	.7763	.2479	.9745
	ZSTS	.4901	.9133	.4625	.1798	.8789
	CusText	.2421	.9291	.7402	.2023	.9092
ChatGPT	.4586	.9822	.5295	<b>.2536</b>	.9226	
TrustPilot <sub>G</sub>	IDT	<b>.4544</b>	<b>.9989</b>	<b>.5444</b>	.2516	.9795
	Adv. Back-translation	.1211	.9854	.8771	.2515	.9665
	Back-translation	.1166	.9873	.8816	.2502	.9624
	TextFooler	.3225	.9913	.6752	.2532	.0195
	TextBugger	.3113	.9932	.6872	.2615	.9692
	BAE	.3215	.9923	.6748	.2555	<b>.9811</b>
	Presidio	.0961	.9985	.9038	.2593	.9768
	ZSTS	.3069	.9526	.6776	.1930	.8755
	CusText	.2325	.9718	.7572	.2322	.9116
ChatGPT	.1683	.9962	.8313	<b>.2748</b>	.9590	
TrustPilot <sub>A</sub>	IDT	<b>.4350</b>	.9946	<b>.5609</b>	.1914	.9726
	Adv. Back-translation	.1563	.9775	.8372	.2225	.9647
	Back-translation	.1230	.9857	.8734	.2227	.9600
	TextFooler	.2818	.9930	.7155	.2233	.9698
	TextBugger	.2711	.9928	.7262	.2388	.9635
	BAE	.3415	.9872	.6537	.2208	.9754
	Presidio	.0507	<b>.9959</b>	.9492	<b>.2565</b>	<b>.9760</b>
	ZSTS	.3064	.9397	.6718	.1750	.8749
	CusText	.1495	.9642	.8441	.2076	.9268
ChatGPT	.3839	.9930	.6147	.2498	.9196	
TOEFL11	IDT	.4485	.9716	.5355	.1028	.9731
	Adv. Back-translation	.5183	.9762	.4689	.1055	.9585
	Back-translation	.4445	.9652	.5363	.1043	.9509
	TextFooler	.4293	.9525	.5444	.1053	.9792
	TextBugger	.4346	.9694	.5488	.1066	.9735
	BAE	.4807	.9542	.4883	.1049	<b>.9860</b>
	Presidio	.3843	.9552	.5895	.0899	.9623
	ZSTS	.4441	.6678	.3229	.0924	.9031
	CusText	.5264	.8658	.3916	.0877	.8868
ChatGPT	<b>.8006</b>	<b>.9878</b>	<b>.1905</b>	<b>.1130</b>	.8799	
Shakespeare	IDT	<b>.6812</b>	<b>.8531</b>	.2250	.0948	.9505
	Adv. Back-translation	.2757	.8492	.6557	.0968	<b>.9642</b>
	Back-translation	.2575	.6654	.5876	.0911	.9402
	TextFooler	.4626	.7149	.3359	.0888	.9496
	TextBugger	.4774	.7511	.3499	.0964	.9606
	BAE	.4256	.6551	.3329	.0880	.9427
	Presidio	.0000	.4927	1.0000	<b>.1365</b>	.9516
	ZSTS	.0754	.3064	.7469	.0832	.8762
	CusText	.2210	.7500	.6791	.0827	.9210
ChatGPT	.5205	.5487	<b>.0472</b>	.1000	.8798	

BAE, which is far from achieving the principal goal), and quite far ahead of adversarial Back-translation’s 0.9134 and ChatGPT’s 0.9122.

IDT stands out in the Shakespeare dataset with the highest RS and a relatively high UR (and third-highest BERTScore). It is effective in both attacking the sentences w.r.t. style and retaining their utility regarding the play they belong to. The number of classes in the utility task is the biggest among all tasks (17). All the other baselines, but remarkably BAE, achieve smaller RS and distort the sentences to the point of losing up to a third of their utility. The extra signal to avoid modifying important tokens for the auxiliary task appears to be important to retain the classification for the utility, especially when considering a large number of classes (17 for Shakespeare).

For TOEFL11, IDT has the fourth largest RS, behind only BAE and adversarial Back-translation, the second biggest UR, and the fourth-highest BERTScore. We note here the potential advantages of Back-translation: Without the signal from the adversary, the RS is (a quite decent) 44.45%, but the signal improves it to 51.83%. However, keeping the stability of the translator is hard, and it is the worst performer in TrustPilot (location), mostly due to mode collapse. The extra signal from the adversary makes the training stability an extra challenge.

In terms of the principal comparator, adversarial Back-translation, the reasons for its relatively poorer performance vary. For TrustPilot<sub>L</sub>, it has trouble retaining accurate predictions on the utility class, which is what brings down its rewrite success there. However, for the others, it has a high UR; the difficulty is in changing the texts sufficiently to obscure the sensitive attribute.

The sanitizer algorithms, namely, CusText, Presidio, and ZSTS, are among the worst performers in general. Presidio aims to modify PII tokens, which may have little effect in classification tasks, whereas ZSTS has the opposite effect: by modifying rare words and thus also harming utility.

IDT also outperforms the TextAttack algorithms (TextFooler, TextBugger, and BAE) across all datasets. These baselines show more stable behavior compared with Back-translation and sanitizers, but they still miss the utility task explicit signal, which makes a difference for IDT.

Regarding ChatGPT, its rewriting success is consistently strong overall, being the second highest in three datasets and the highest in one. ChatGPT was used as a zero-shot rewriter, but its performance is still behind IDT. The latter requires fine-tuned models, which appears to be an advantage over the zero-shot approach. Interestingly, the highest performance is for TOEFL11, which consists of the longest texts; adversarial Back-translation also performs strongly here.

We observe that USE cosine similarity differs quite noticeably from our primary fine-grained utility metric BERTScore. By USE, ChatGPT often scores well (highest in three of the dataset tasks), even when it is low by BERTScore: For instance, on TOEFL11 it has the highest USE score and lowest BERTScore. We reflect on this in Section 6.3.

Details on time taken to generate attacks is in Appendix G. Details on number of successful attacks are in Appendix H. We show the rewrite success in a scatter plot in Appendix Figure H.1.

### 6.3 Digging into Reformulated Texts

We show reformulated texts generated by the methods of Table 4, alongside their labels.

For the first TrustPilot example, rewriting methods mostly chose one or two words to change, with ZSTS reformulating very extensively. The second TrustPilot example requires slightly more rewriting, with some further differentiation among them

**Table 4**

Examples of reformulated sentences generated by the methods. The top two sets of samples come from the TrustPilot dataset; the third set is from TOEFL11 and the bottom set is from Shakespeare sentences.

Rewrite Method	Text
–	I would certainly recommend parcel hero and use their services again
IDT	I would certainly recommend parcel <b>entertainer</b> and use their services again
TF	I would certainly <b>commended</b> parcel hero and use their services again
Adv. Back-trans.	i would <b>definitely</b> recommend parcel hero and use their services again
Back-trans.	I would certainly recommend <b>a package</b> hero and use their services again.
TB	I would certainly <b>reCommend</b> parcel hero and use their services again
BAE	I would certainly <b>marry</b> parcel hero and use their services again
CusText	I would <b>yet</b> recommend parcel <b>fame</b> and use their services <b>leaving</b>
Presidio	I would certainly recommend parcel hero and use their services again
ZSTS	i would <b>simply join some deliveries not use these weapons</b> -
ChatGPT	I would <b>definitely</b> recommend Parcel Hero and <b>would</b> use their services again.
–	The flowers my sister received were presented beautifully she said and were in lovely condition.
IDT	The flowers my <b>wife</b> received were presented beautifully she said and were in <b>lush</b> condition.
TF	The flowers my <b>hermano</b> received were presented <b>phenomenally</b> she said and were in <b>belle</b> condition.
Adv. Back-trans.	<b>the</b> flowers my sister received were <b>beautifully presented</b> she said and were in <b>beautiful</b> condition.
Back-trans.	The flowers my sister received were beautifully presented she said and were in <b>very good</b> condition.
TB	The flowers my <b>sibling</b> received were presented beautifully she said and were in <b>ravishing</b> condition.
BAE	<b>he</b> flowers my sister received were presented beautifully she said and were in <b>decent</b> condition.
CusText	The flowers <b>your daughters</b> received were <b>previously</b> beautifully <b>himself afp with they</b> in lovely condition.
Presidio	The flowers my sister received were presented beautifully she said and were in lovely condition.
ZSTS	<b>the items</b> my family said were <b>cut by</b> she said <b>them</b> were in <b>superb</b> condition .
ChatGPT	The flowers my <b>brother</b> received were presented beautifully, <b>he mentioned</b> , and were in lovely condition.
–	Of course, the fact will be useless when the main idea is not transported to the students .
IDT	Of course, the fact will be useless when the main idea is not <b>traced</b> to the students .
Adv. Back-trans.	of course, the fact will be useless when the main idea is not transported to students.
Back-trans.	Of course, the fact will be useless when the main idea is not carried to the students.
TF	<b>With</b> course, the fact will be useless when the main idea is not transported to the students .
TB	<b>Del</b> course, the fact will be useless when the main idea is not transported to the students .
BAE	of course, the fact will be useless when the main idea is not transported to the students .
CusText	Of course <b>with the what</b> will be useless <b>again</b> the main idea is if transported <b>its</b> the students <b>for</b>
Presidio	Of course, the fact will be useless when the main idea is not transported to the students .
ZSTS	<b>of</b> course, <b>any solution</b> will be <b>reached so</b> the <b>good system</b> is not <b>visible to the lab</b> .
ChatGPT	Certainly, facts lose their significance if the core concepts are not conveyed to the students. It is essential for learners to grasp the underlying ideas rather than just memorizing isolated data points.
–	no more than my staying here in rome might mean to you in egypt.
IDT	no more than my staying here in rome might mean <b>up</b> you in egypt.
Adv. Back-trans.	no more than my <b>stay</b> here in rome might mean to you in egypt.
Back-trans.	No more than my <b>stay</b> here in Rome could mean for you in Egypt.
TF	no more than my staying here in rome might <b>intentioned</b> to you in egypt.
TB	no more than my staying here in rome might <b>meaning</b> to you in egypt.
BAE	no more than my staying here in rome might <b>render</b> to you in egypt.
CusText	<b>nor</b> more <b>less something</b> staying <b>close on</b> rome might <b>exactly</b> to you in egypt.
Presidio	no more than my staying here in rome might mean to you in egypt.
ZSTS	no more <b>everything you living</b> here in <b>greece must</b> mean <b>having study</b> in <b>fact</b> .
ChatGPT	No <b>longer doth</b> my <b>sojourn</b> here in Rome <b>bear weight unto thee</b> in Egypt's embrace.

apparent: Whereas IDT, the backtranslation methods, the adversarial attack methods, and ChatGPT required two or three words changed, the sanitization methods all showed extensive changes.

For TOEFL11, the original sentence, written by a Chinese learner, was modified in ways to deceive the target model to classify it as written either by a French or an Arabic

speaker. The topic, “if it is more important for students to understand ideas and concepts than it is for them to learn facts,” shortened in the table as “Students & learning,” was kept. Here especially, ChatGPT is more aggressive in its reformulations. For the TOEFL11 sentence, it is quite verbose and turns the sentence into a double-sentence text that is quite distant from the original. Likewise, in the Shakespeare sentence, most words are replaced.

The examples also highlight some of the limitations of IDT, which it shares with other methods in some cases. In the first example, the proper noun phrase “Parcel Hero” is changed to “Parcel Entertainer” (with the backtranslation method and CusText similarly changing this). In the IDT approach, this could potentially be handled with constraints that, for instance, prevent named entities being changed, but we do not consider those sorts of constraints here.

Returning again to the question of differences between BERTScore and USE cosine similarity, we note that the extensive rewriting by ChatGPT on TOEFL11 is precisely where ChatGPT performed best by USE and worst by BERTScore. It may be the case that for these kinds of rewriting, USE captures a looser notion of semantics and BERTScore the intended stricter notion.<sup>15</sup>

Table 5 shows measurements of difference between the original sentences and the rewritten ones. The key takeaway is that for IDT the rewritten sentences were close to the original, and comparable to the (single-task) adversarial attacks. In general, adversarial texts generated by TextBugger present the results at least orthographically closest to the original. However, this algorithm had the poorest performance among all attacks, which is an indication that its changes were too subtle. This is corroborated by our grammar checker, which deems TextBugger texts as the most correct in general—and TextFooler produces the smallest number of correct texts for all datasets. IDT holds the highest Jaccard similarity (which also correlates well with the percentage of perturbed words) between adversarial and original texts in four out of the five datasets, on top of also being the attack with the highest RS in four datasets.

We notice that the adversarial training of Xu et al. (2019) is heavily inspired on generative adversarial networks (Goodfellow et al. 2014), which commonly face instability training issues. While training our adversarial Back-translation, we have encountered mode collapse, as well as memory issues while training, since two neural networks must be held in memory. We found it particularly challenging to find a balance in some cases, as in TrustPilot (location), for which mode collapse produced the poorest quality metrics.

We also notice the importance of the constraint of ensuring the same POS tags between original and adversarial tokens. The baselines, which do not enforce this strict constraint, generate around half of the texts with different part-of-speech. The cosine similarity constraint requirements imposed by the baselines is not enough to prevent mismatching POS tags.

Last, we bring in Table 6 a few practical examples with the respective scores predicted by the classifier to better understand individual-level utility trade-offs: In some cases, greater stylistic rewritings lead to lower BERTScores and more privacy classification changes, while subtler modifications tend to maintain high semantic alignment but may sometimes not fully achieve the privacy alteration.

Take, for example, the first example in Table 6, from the TOEFL dataset: The sentence has only one token replaced (*traced* for *transported*). The privacy class has been

---

<sup>15</sup> This would be an interesting topic for future study.

**Table 5**

Measurements of difference between reformulated texts and the original ones. The scale is in the range of 0 to 1, with the exception of changed words where the ratio can be  $>1.0000$  for long reformulated sentences.

Dataset	Rewrite Method	Matching POS tags	Grammar correct	Levenshtein ratio	Changed words	Jaccard sim.	Meteor	BLEU	RougeL
TrustPilot <sub>L</sub>	IDT	<b>1.0000</b>	.7222	.9614	.0570	.8945	.9440	.9189	.9439
	Adv. Back-trans.	.0000	.0019	.1803	$>1.0000$	.0049	.0135	.0000	.0034
	Back-trans.	.0237	.7782	.8591	.7056	.6039	.8143	.4779	.7952
	TextFooler	.5776	.6794	.9402	.0823	.8466	.9232	.8716	.9193
	TextBugger	.6575	.8200	<b>.9827</b>	.0989	.8896	.9483	.9066	.9388
	BAE	.6460	.7566	.9490	.0706	.8632	.9332	.8887	.9292
	CusText	.0030	.0251	.7800	.5147	.4866	.6079	.3654	.6301
	Presidio	.0000	.8157	.9567	<b>.0331</b>	<b>.9387</b>	<b>.9615</b>	<b>.9208</b>	<b>.9547</b>
	ZSTS	.0078	.1968	.6507	.7486	.2943	.5255	.3285	.5413
ChatGPT	.0000	<b>1.0000</b>	.6251	$>1.0000$	.2638	.5194	.1671	.4789	
TrustPilot <sub>G</sub>	IDT	<b>1.0000</b>	.7039	.9592	<b>.0549</b>	.8952	.9450	<b>.9135</b>	.9447
	Adv. Back-trans.	.0351	.8694	.8644	.6611	.6121	.8307	.5122	.8317
	Back-trans.	.0343	.7974	.8499	.7033	.6256	.8019	.5409	.7835
	TextFooler	.5854	.6778	.9395	.0843	.8424	.0702	.8745	.9168
	TextBugger	.4351	.7650	<b>.9689</b>	.1470	.8446	.9262	.8716	.9145
	BAE	.5850	.7580	.9483	.0766	.8556	.9293	.8721	.9269
	CusText	.0109	.0350	.7982	.4836	.5194	.6470	.4001	.6627
	Presidio	.0014	.8809	.9569	.1107	<b>.9309</b>	<b>.9598</b>	.9110	<b>.9595</b>
	ZSTS	.0017	.1783	.6341	.7646	.2613	.5241	.2442	.5408
ChatGPT	.0346	<b>.9939</b>	.8206	.7871	.5444	.7702	.5134	.7460	
TrustPilot <sub>A</sub>	IDT	<b>1.0000</b>	.6787	.9378	<b>.0930</b>	.8461	.8999	<b>.9158</b>	.9098
	Adv. Back-trans.	.0514	.8776	.8473	.6671	.5983	.7994	.4586	.8148
	Back-trans.	.0438	.8043	.8377	.7932	.6077	.7809	.4912	.7680
	TextFooler	.5234	.6308	.9180	.1111	.8009	.8934	.8609	.8907
	TextBugger	.3411	.7357	<b>.9583</b>	.2092	.8089	.9048	.8528	.8930
	BAE	.5439	.7396	.9284	.1014	.8214	.8982	.8754	.9012
	CusText	.0227	.0417	.7949	.5137	.5180	.6403	.3995	.6602
	Presidio	.0010	.8864	.9530	.1261	<b>.9295</b>	<b>.9575</b>	.9143	<b>.9554</b>
	ZSTS	.0063	.2049	.6304	.7907	.2578	.5147	.2576	.5338
ChatGPT	.0010	<b>.9984</b>	.5800	$>1.0000$	.2429	.4600	.1605	.4193	
TOEFL11	IDT	<b>1.0000</b>	.4756	.9467	.0755	.8739	.9310	.8532	.9257
	Adv. Back-trans.	.0755	.8652	.8835	.6396	.5013	.8404	.5554	.8130
	Back-trans.	.0600	.8452	.8514	.6711	.4882	.7916	.4961	.7511
	TextFooler	.7839	.4565	.9438	.0801	.8582	.9349	.8582	.9201
	TextBugger	.5237	.5535	<b>.9758</b>	.1026	.8717	.9427	.8756	.9218
	BAE	.8565	.5898	.9557	<b>.0722</b>	.8733	<b>.9473</b>	<b>.8778</b>	.9373
	CusText	.0021	.0081	.7340	.5934	.4298	.5434	.2327	.5651
	Presidio	.0037	.6455	.9184	.1609	<b>.8871</b>	.9276	.8113	<b>.9332</b>
	ZSTS	.0443	.3513	.6781	.5670	.3959	.5734	.2934	.5775
ChatGPT	.0001	<b>.9998</b>	.4018	$>1.0000$	.1327	.2950	.0273	.2169	
Shakespeare	IDT	<b>1.0000</b>	.4188	.8822	.1872	.7015	.7846	.7579	.8160
	Adv. Back-trans.	.1130	.7470	.8928	.5535	.6514	.7961	.6804	.8136
	Back-trans.	.1087	.7484	.7727	.8460	.3816	.7001	.3710	.6773
	TextFooler	.5656	.3744	.8534	.2069	.6750	.7769	.7032	.7992
	TextBugger	.5505	.6221	<b>.9422</b>	.1747	.7530	.8599	.7869	.8524
	BAE	.5505	.5589	.8263	.2408	.6343	.7241	.6879	.7615
	CusText	.0689	.0923	.7789	.4094	.4911	.5807	.3832	.6479
	Presidio	.0000	.7536	.8649	<b>.0993</b>	<b>.8239</b>	<b>.9103</b>	<b>.7925</b>	<b>.9068</b>
	ZSTS	.0975	.3963	.6046	.7512	.2194	.4782	.2294	.4724
ChatGPT	.0038	<b>.9519</b>	.4437	$>1.0000$	.1136	.3480	.0252	.2773	

changed (from Chinese to French), while the BERTScore indicates that, in terms of fine-grained semantics, the rewritten sentence is similar to the original. In contrast, the second example (also from TOEFL), while also successfully changing the privacy class, has much greater changes and consequently lower similarity. This is perhaps not surprising here, as the prediction of Hindi for the original has what can be interpreted as a much higher level of confidence (relative to the Chinese case of the first example),

**Table 6**

Examples of reformulations for Shakespeare TOEFL (top two) and TrustPilot (remainder) alongside the score of the predicted class for the privacy task, and the BERTScore between the two sentences. The last example has some of the text elided.

Source	Text	Privacy prediction (score)	BERTScore
Original	Of course, the fact will be useless when the main idea is not transported to the students .	Chinese (.1868)	.9798
IDT	Of course, the fact will be useless when the main idea is not traced to the students .	French (.2019)	
Original	In it 's advertisement they claim that they provide hassle free and customer oriented service in their outlets .	Hindi (.7835)	.8628
IDT	In it 'comes adverticomeselement they claim that they provide hacomeselesle free and cucomestomer oriented comeseervice in their outletcomes .	Arabic (.7511)	
Original	Ordered one day delivered 2 days later, excellent! Will be happy to buy again.	Female (.6303)	.9733
IDT	Ordered one day touched 2 days later, excellent! Will be happy to buy again.	Female (.5361)	
Original	The flowers my sister received were presented beautifully she said and were in lovely condition. I would recommend this company.	Female (.9697)	.9890
IDT	The flowers my wife received were presented beautifully she said and were in lush condition. I would recommend this company.	Male (.9859)	
Original	Ordered items and they arrived. Cannot say anymore than that. I would definitely use them again for ordering items for my cats.	Female (.9054)	.9883
IDT	Ordered items and they arrived. Cannot say anymore than that. I would definitely use them again for ordering items for my dudes.	Male (.9786)	
Original	to update my recent, rather negative review (which i didn't realise would go live!) by saying that i had a call immediately from the drink shop after posting this review and they are fully refunding my delivery charge and were extremely apologetic and helpful. full marks for brilliant customer service, very, very impressive! as always, the items arrived - however, the packaging was really badly damaged! [...] i believe you are still with them, which supprises me. i do want to order from you again but i would expect better value for the money i'm paying for delivery.	UK (.4950)	.9943
IDT	to update my recent, rather negative review (which i didn't realise would go live!) by saying that i had a call immediately from the drink shop after posting this review and they are fully refunding my tender charge and were extremely apologetic and helpful. full marks for brilliant customer service, very, very impressive! as always, the items arrived - however, the packaging was really badly damaged! [...] i believe you are still with them, which supprises me. i do want to order from you again but i would expect better value for the money i'm paying for tender.	France (.4041)	

and the prediction of Arabic for the perturbed text is also of high confidence, which could really only be the result of major changes.

The remaining examples are in the domain of social media posts, like our motivating scenario, and also show a range of outcomes. The third example, like the first, only changes one word (*touched* for *delivered*), and so likewise has a high BERTScore, but is not quite enough of a perturbation to flip the privacy class. The fourth example is the

one used as an illustration in Figure 1: It also only has a fairly minimal perturbation, with two words changed (*wife* for *sister* and *lush* for *lovely*), but the very strong most likely inference from *wife* strongly flips the predicted privacy class. A similar but less obvious example occurs in the fifth example, where *cats* is replaced by *dudes*. And in the final example, of a very long text where the only word replaced is *delivery*, by *tender* (albeit multiple times), the privacy class is changed while the text remains very close.

#### 6.4 Ablation and Further Studies

*Different Target Model.* We study how transferable IDT and TextAttack baselines are when the auxiliary models have a distinct architecture from the target model. This only applies to those of our rewriting approaches that use auxiliary models. The target is a distilled GPT2, whereas the auxiliaries are distilled RoBERTa. We also report the differences between the RS and the UR reported in Table 3 in Table 7.

IDT is the clear winner when this kind of transferability is taken into account, and the baselines are much more reliant on the architecture to find words to substitute. IDT achieves the highest RS in all datasets but TrustPilot<sub>G</sub>, which ranks second by a tiny margin of 0.15% behind BAE.

*ChatGPT as Target Model.* We consider how robust the IDT rewritings are to inference by other models. For this, we take the sentences reformulated by IDT and ask ChatGPT (as the best of the LLM inference models from Section 6.1) to classify them in the privacy task for each dataset. Our goal is to assess how IDT reformulations conceal privacy

**Table 7**

RS and UR for target and auxiliary models with different architectures, alongside their differences to the setting reported in Table 3. The scale is in the range of 0 to 1.

Dataset	Rewrite	RS	Diff.	UR	Diff.
TrustPilot <sub>L</sub>	IDT	<b>.3947</b>	-.2219	<b>1.0000</b>	.0000
	TextFooler	.3608	-.0819	.9877	.0005
	TextBugger	.3663	-.1712	.9969	.0094
	BAE	.3023	-.1047	.9941	.0074
TrustPilot <sub>G</sub>	IDT	.2623	-.1921	<b>1.0000</b>	.0011
	TextFooler	.2415	-.0810	.9927	.0014
	TextBugger	.2076	-.1037	.9941	.0009
	BAE	<b>.2638</b>	-.0577	.9915	-.0008
TrustPilot <sub>A</sub>	IDT	<b>.3758</b>	-.0592	<b>.9957</b>	.0011
	TextFooler	.2844	.0026	.9930	.0000
	TextBugger	.2737	.0026	.9928	.0000
	BAE	.3510	.0095	.9892	.0020
TOEFL11	IDT	<b>.4210</b>	-.0275	<b>.9845</b>	.0129
	TextFooler	.3589	-.0704	.9549	.0024
	TextBugger	.3702	-.0644	.9691	-.0003
	BAE	.3843	-.0964	.9559	.0017
Shakespeare	IDT	<b>.4022</b>	-.2790	<b>.7873</b>	-.0658
	TextFooler	.3519	-.1107	.7111	-.0038
	TextBugger	.3245	-.1529	.7327	-.0184
	BAE	.3539	-.0717	.6234	-.0317

**Table 8**

ChatGPT inference on sentences reformulated by IDT. Accuracy for the privacy tasks. The scale is in the range of 0 to 1.

Dataset	ChatGPT acc.
TrustPilot <sub>L</sub>	.0611
TrustPilot <sub>G</sub>	.5846
TrustPilot <sub>A</sub>	.5475
TOEFL11	.2103
Shakespeare	.3875

information from ChatGPT’s inference. Ideally, the data holder wishes the accuracy of the LLM to be as low as possible.

The accuracy of ChatGPT in inferring the actual label of sentences rewritten by IDT is shown in Table 8. Compared to accuracy over original sentences in Table 2, we see that IDT manages to hide the privacy attributes from ChatGPT, since it struggles to guess the original class of the sentences.

*Relaxing Constraints.* We examined relaxing the constraint of forcing adversarial texts to have the same POS tags for every token in the original text. More specifically, we redesign IDT to allow nouns to be swapped by verbs and vice-versa, as in the TextAttack implementation of TextFooler and BAE. As in TextAttack, we also use the “universal” tagset of NLTK library (Bird and Loper 2004) to compute the POS tags. Table 9 shows the results.

Except for TOEFL11, relaxing the constraints helps the RS for all datasets. For Shakespeare, it is particularly helpful to make the RS above 70%, the highest results across all our experiments.

*Ignoring Important Tokens for Utility.* We modify IDT to ignore all words for the utility task, to make the rewriting behavior akin to the adversarial attacks from the literature. Then, we measure the effects of utility retention and the predictions for the privacy tasks. Results are shown in Table 9.

**Table 9**

RS and UR for IDT under different constraints, alongside their differences to the setting reported in Table 3. IDT<sub>s</sub> means verbs can be swapped by nouns and vice-versa, and IDT<sub>u</sub> ignores the utility task to sample replacement words. The scale is in the range of 0 to 1.

Dataset	Attack	RS	Diff.	UR	Diff.
TrustPilot <sub>L</sub>	IDT <sub>s</sub>	.5966	−.0200	1.0000	.0000
	IDT <sub>u</sub>	.6067	−.0099	1.0000	.0000
TrustPilot <sub>G</sub>	IDT <sub>s</sub>	.4409	.0135	.9978	−.0011
	IDT <sub>u</sub>	.4305	−.0239	.9957	−.0032
TrustPilot <sub>A</sub>	IDT <sub>s</sub>	.4650	.0300	.9986	.0040
	IDT <sub>u</sub>	.4304	−.0046	.9933	−.0013
TOEFL11	IDT <sub>s</sub>	.4293	−.0192	.9662	−.0054
	IDT <sub>u</sub>	.4342	−.0143	.9578	.0036
Shakespeare	IDT <sub>s</sub>	.7177	.0365	.8742	.0211
	IDT <sub>u</sub>	.5476	−.1336	.7574	−.0957

We see that such relaxation yields a negative impact for all datasets in terms of RS. The sharpest decrease happens for Shakespeare sentences.

## 6.5 Human Evaluation Results

We report the human evaluation regarding utility retention and grammaticality and fluency judgments. In this, we just compare our model IDT with the principal baseline, adversarial backtranslation, and the TextFooler adversarial attack that is at the base of IDT.

*Utility Evaluation.* We asked annotators to classify the sentences with respect to the utility task. For TOEFL11, it means judging whether the sentence talks about “cars” or “students & learning,” and for TrustPilot it means rating it as a “good” or “bad” review. We report the accuracy of the annotators in Table 10.

In general, humans can discern with ease the class of the sentences for all algorithms. Numerically, though, IDT achieves the highest utility accuracy score for the TOEFL11 dataset, with a score of 95.33%. This surpasses even the original sentences (93.33%) and other transformation methods like TextFooler and adversarial Backtranslation (both at 94.66%).

*Grammar and Fluency.* The second assignment asked annotators to judge the sentences with respect to their grammar and fluency. Their judgments are summarized in Table 11.

For all algorithms, as well as the original sentences, across both datasets, scores are mostly between 3 and 4 in our 1–5 Likert scale, with a couple above 4 (higher scores being better). A score of 3 indicates basically fluent and understandable, and a score of 4 that there are only one or two minor errors.

We notice that all algorithms achieve better grades in TrustPilot than TOEFL11. This is explained by the fact that TOEFL11 sentences are written by learners of English and taken from long essays. Therefore TOEFL11 sentences are more prone to contain errors than TrustPilot reviews, although the original sentences here also score below 5 (free of grammatical errors) as these are often quickly written product reviews that

**Table 10**

Human evaluation results for topic (TOEFL11) and sentiment (TrustPilot) classification.

Dataset	Utility accuracy (%)			
	Original	TextFooler	Adv. Back-trans.	IDT
TOEFL11	93.33	94.66	94.66	95.33
TrustPilot	100.00	96.90	92.13	96.11

**Table 11**

Human evaluation results for grammar and fluency. The closer to 5, the better quality the sentences hold according to the annotators.

Dataset	Grammar and fluency (1–5)			
	Original	TextFooler	Adv. Back-trans.	IDT
TOEFL11	3.91	3.59	3.98	3.36
TrustPilot	4.16	3.84	4.26	3.80

**Table 12**

TOEFL (top) and TrustPilot (bottom) texts (original and IDT version) with three annotator scores.

Rewrite	Text	Annotators
–	Old people enjoy life not less than young people, but i think that they have already left this important moment of their life and they prefer to stay togheter in quite and not busy place .	3/4/5
IDT	Old people enjoy life not less than young people, but i think that they <b>over</b> already left this important moment of their life and they prefer to stay togheter in quite and not busy place .	3/3/3
–	i called printerland for advice, as i was torn between two different mono printers. i had a gut feeling for which to go with, and luckily the person i spoke to was able to back up what i thought with some extra information. i eventually opted for an oki b710, having compared prices elsewhere and found printerland the cheapest. the printer arrived by 11am the following morning! i couldn't have been happier with the advice that was offered on the phone, or the service received. the printer has been great so far too! we will definitely use printerland again.	3/4/5
IDT	i called printerland for advice, as i was torn between two different mono printers. i had a gut feeling for which to go with, and luckily the person i spoke to was able to back up what i thought with some extra information. i eventually opted for an oki b710, having compared prices elsewhere and found printerland the cheapest. the printer arrived by 11am the following morning! i couldn't have been happier with the advice that was offered on the phone, or the service received. the printer has been great so far too! we will definitely use printerland again.	3/3/3

have not been proofread. In terms of methods, the adversarial Back-translator clearly produces the most fluent texts. Interestingly, in fact, the adversarial Back-translator achieves better scores than the original sentences in both datasets perhaps because, as a generative model, it is fixing spelling mistakes by learners. This doesn't happen with the other methods, which are token-based replacements. And although outperformed on this property by the adversarial Back-translator, the IDT scores still indicate an acceptable degree of fluency.

We give two examples in Table 12 of the human evaluation of an original text and the IDT rewriting, for TOEFL and for TrustPilot. In both cases, the average score is 4 for the original text, but 3 for the IDT rewriting. In the TOEFL example, the original has several errors, and the IDT rewriting differs only in one word. In the TrustPilot example, the only differences are in punctuation. In both of these, the loss of fluency as scored by annotators does not (much) impair the ability to interpret the text, and so may be an acceptable price for a user wishing to trade off for more privacy.

### 7. Conclusions and Future Work

Inference of the private attributes of writers, such as age or native language, are detectable with increasing ease by machine learning models in scenarios where text is made public, such as product reviews or social media forums. In this article, we have proposed a method based on adversarial attacks for rewriting text to prevent detection of such private attributes, while maintaining the utility with respect to some primary task (e.g., sentiment classification).

We have shown that most existing privacy-protecting methods that rewrite text, such as text sanitization, do not already solve this specific problem. We have further shown that IDT performs the best against several baselines, including generative models with the same goal, in terms of ability to preserve privacy while maintaining utility; and it manages to keep texts fairly similar to the original while doing so. This makes it a strong baseline for future work, in particular on shorter texts, on a task that we consider quite important.

There are several directions for future work. IDT, or approaches similar to it, could be improved by starting from a different adversarial attack, using different constraints, applying different methods for identifying words to be changed, and so on. In particular, constraints reflecting fine-grained utility—for instance, in terms of semantic similarity metrics—would be a high priority to investigate. These would need to consider issues raised by Morris et al. (2020a) in adversarial attacks, regarding the need for metrics and thresholds to be carefully selected to calibrate with human judgments gathered via a careful framing. Other constraints to prevent the changing of named entities, as noted in the discussions of examples, could help improve quality. Further, analyzing and strengthening the generalizability of the method against a range of inference attackers, for instance, by including several diverse types of auxiliary models, would be important for any real-world deployment, where the attackers could be unknown or of multiple types. Another particular aspect that could be a focus of improvement is in human-judged fluency, where the generative baseline, adversarial Back-translation, was clearly superior. Generative methods that can avoid problems like mode collapse are thus also an important future direction.

## A. Appendix A: Annotation Guidelines

*Utility retention.* What is this sentence talking about?

«Sentence»

- Students & Learning.
- Cars.

*Grammar and fluency.* Please judge the text according to its grammar and fluency.

1. Not in the form of human language.
2. Cannot understand what is the meaning, but is still in the form of human language.
3. Basically fluent and has three or more minor grammatical errors or one serious grammatical error that does not have strong impact on understanding.
4. Fluent and has one or two minor grammatical errors that do not affect understanding.
5. Without any grammatical error.

**Table A.1**

Data splits for target (left) and auxiliary models (right). TrustPilot<sub>L, G, A</sub> refer to Location, Gender, and Age views of this dataset.

Dataset	Train	Valid.	Test	Train	Valid.
TrustPilot <sub>L</sub>	5,960	663	1,000	5,959	663
TrustPilot <sub>G</sub>	100,000	1,000	5,000	100,000	1,000
TrustPilot <sub>A</sub>	85,745	1,000	5,000	85,745	1,000
TOEFL11 <sub>S</sub>	72,301	16,146	15,951	65,070	7,231
TOEFL11 <sub>C</sub>	14,214	3,182	3,105	12,793	1,422
Shakespeare	23,491	2,611	4,000	23,491	2,611

## B. Appendix B: Data Splits

We split each dataset into target and auxiliary model data as shown in Table A.1. TOEFL11<sub>S</sub> refers to the TOEFL11 experiments described in the main body of this article, and TOEFL11<sub>C</sub> is about the extra experiments from Appendix D.

## C. Appendix C: Implementing the Adversarial Back-Translation Baseline

We reimplemented the Back-translation with adversarial training proposed by Xu et al. (2019), with a few changes since more powerful architectures have been developed since then.

Their framework involves three steps: (i) using two translators (English to French, followed by French to English) to turn the datasets labeled with sensitive attributes into a parallel corpus; (ii) train a classifier and a back-translator in an adversarial fashion using the parallel corpus from the step before; (iii) use the back-translator to generate adversarial sentences.

In their study, Xu et al. (2019) trained a translator for step (i) using the Europarl v7 dataset from Koehn (2005). The authors reported a BLEU score of 36.24%. We used the English to French and French to English models provided by Tiedemann and Thottingal (2020) to translate the texts from our datasets. As explained in Section 5, we used two MarianMT pretrained models (Tiedemann and Thottingal 2020) to generate the parallel corpus.

For step (ii), the authors trained the Transformer from Vaswani et al. (2017). We instead fine-tuned a pretrained BART (Lewis et al. 2020). For the first 10 epochs we only trained the adversary classifier (a BART classification head). Then, for the next 10 epochs, we only fine-tuned the translator. Finally, we trained both for 14 epochs. Our learning rate was set to 1e-5, batch size of 32 (except for TOEFL, for which we used 16) and Adam optimizer for both translator and adversary. Xu et al. (2019) reported they used the mean of the hidden representations of the Transformer as input to the classifier. We found it more useful to take the last hidden state of BART.

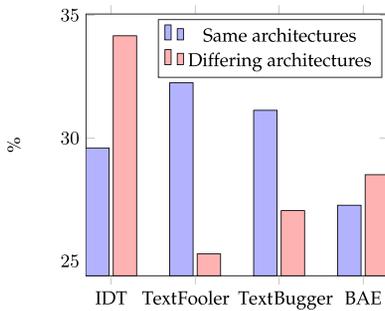
## D. Appendix D: Experiments with Longer Texts

Since the TOEFL11 dataset contains longer essays than the maximum input size of base Transformer models (usually 512 or 1024), we organized an extra set of experiments for which we split the texts into chunks of 5 sentences. Hereafter, we call this view of the TOEFL11 as TOEFL11<sub>C</sub>, while the other one, described in the main body of the article,

**Table D.1**

Accuracy of the target and auxiliary models on the test sets of TOEFL11<sub>C</sub>. “Both” means samples were correctly classified in both tasks.

Dataset	Model	Privacy	Utility	Both
TOEFL11 <sub>C</sub>	Target	.5574	.9742	.5443
	Aux.	.5539	.9784	.5407

**Figure D.1**

Rewrite Success rates against TOEFL11<sub>C</sub>.

**Table D.2**

Measurements on the quality of adversarial texts against the original ones from TOEFL11<sub>C</sub>.

Dataset	Attack	Matching POS tags	Grammar correct	Cosine sim.	Levenshtein ratio	% Perturbed words	Jaccard sim.	Meteor	BERTScore F1
TOEFL11 <sub>C</sub>	IDT	1.0	.4587	.2562	.9808	.0261	.9477	.9741	.9857
	TextFooler	.5849	.4174	.2595	.9724	.0348	.9145	.9688	.9864
	TextBugger	.3977	.4981	.2598	.9891	.0988	.9249	.9733	.9821
	BAE	.6320	.4528	.2614	.9745	.0375	.9137	.9696	.9882

we refer to now as TOEFL11<sub>S</sub>. Accuracy of the target and auxiliary models on the test sets are depicted in Table D.1.

IDT appears to have more difficulty in dealing with the longer texts from TOEFL11<sub>C</sub>, made of chunks of sentences, than the single sentences of TOEFL11<sub>S</sub>. This phenomenon is also observed for all baselines.

Figure D.1 shows the Rewrite Success rates when target and auxiliary models have the same (blue) and distinct (pink) architectures. Under the first setting, IDT is outperformed by TextFooler and TextBugger, but it stays ahead of BAE. However, IDT achieves the biggest RS when architectures differ, showing stronger transferability.

Regarding the quality of the adversarial texts, Table D.2 shows that IDT performs better in qualitative terms: It keeps the largest Jaccard similarity, and now it also has the least percentage of perturbed words and biggest Meteor, surpassing BAE from Table 5.

## E. Appendix E: ChatGPT Prompts

We show in Table E.1 and Table E.2 the system prompts used to instruct ChatGPT to classify sentences and to reformulate sentences, respectively. The TrustPilot privacy

**Table E.1**

Prompts used for both privacy and utility tasks when prompting ChatGPT to act as the target model in classification tasks.

Dataset and Task	Prompt
Shakespeare, privacy	Shakespeare wrote many plays in the sixteenth and seventeenth centuries. English writing style then, in terms of vocabulary and grammar and other aspects, was different from now. You need to classify a Shakespeare sentence regarding its writing style. The style can be either Modern or Old. Classify the sentence as either 'old' or 'modern'. Your return is a valid JSON response with a single key called 'class' and a value of either 'old' or 'modern'.
Shakespeare, utility	Shakespeare wrote many plays in the sixteenth and seventeenth centuries. You need to classify which play a Shakespeare sentence belongs to. It can be from one of the following plays: 'muchado', 'henryv', 'midsummer-nights-dream', 'macbeth', 'hamlet', 'richardiii', 'othello', 'merchant-of-venice', 'tempest', 'romeojuliet', 'antony-and-cleopatra', 'lear', 'juliuscaesar', 'errors', 'asyoulikeit', 'twelfthnight', or 'shrew'. [Write these options out as proper words.] Your return is a valid JSON response with a single key called 'class' and a value of one of those plays.
TOEFL11, privacy	When people grow up speaking one language (their native language) but learn another later, their writing can leave clues to their native language in vocabulary, grammar, or other aspects of style. You need to guess the native language of a writer based on an English text (note that the text is in English, but the native language of its author is not English). The native language of the author can be one of the following eleven options: ARA, DEU, ZHO, FRA, ITA, SPA, TUR, HIN, JPN, KOR, TEL. [Write these out, e.g. TEL as Telugu.] Guess the native language based on the writing style of the text. Your return is a valid JSON response with a single key called 'class' and a value of one of the eleven options.
TOEFL11, utility	You need to classify the topic of a text written students. There are eight topics, and they are: ('P1': 'It is better to have broad knowledge of many academic subjects than to specialize in one specific subject.'; 'P2': 'Young people enjoy life more than older people do.'; 'P3': 'Young people nowadays do not give enough time to helping their communities.'; 'P4': 'Most advertisements make products seem much better than they really are.'; 'P5': 'In twenty years, there will be fewer cars in use than there are today.'; 'P6': 'The best way to travel is in a group led by a tour guide.'; 'P7': 'It is more important for students to understand ideas and concepts than it is for them to learn facts.'; 'P8': 'Successful people try new things and take risks rather than only doing what they already know how to do well.'). Classify the text as one of these eight topics: P1, P2, P3, P4, P5, P6, P7, or P8. Your return is a valid JSON response with a single key called 'class' and a value with the topic.
TrustPilot <sub>L</sub> , privacy	You need to guess the country of a person based on a product review they have written. Your return is a valid JSON response with a single key called 'class' and a value of either 'fra', 'uk', 'den', 'ger', 'us'.
TrustPilot <sub>G</sub> , privacy	You need to guess the gender of a person based on a product review they have written. Your return is a valid JSON response with a single key called 'class' and a value of either 'Male' or 'Female'.
TrustPilot <sub>A</sub> , privacy	You need to guess the age of a person based on a product review they have written. Your return is a valid JSON response with a single key called 'class' and a value of either 'Young' or 'Elder'.
TrustPilot <sub>L,G,A</sub> , utility	You need rate a product review. Your return is a valid JSON response with a single key called 'class' and a value of either 'good review', 'bad review' or 'neutral review'.

and utility attributes were similar to those investigated in Staab et al. (2024a), so the prompts we used were based on their minimalist, successful prompts for their LLMs. For the other two datasets, to our knowledge there has been no similar work, so we experimented with a variety of prompts. The ones with more context proved better. (Token limits precluded still longer prompts.)

**Table E.2**

Prompts used for prompting ChatGPT to rewrite sentences.

Dataset	Prompt
Shakespeare	You need to rewrite a Shakespeare text. Turn its style different from row['privacy'], but keep it looking as a row['utility'] play.
TOEFL11	You need to rewrite a text written by TOEFL students. Turn the native language of the author different from row['privacy'], but keep it looking as an text about row['utility'].
TrustPilot <sub>L</sub>	You need to rewrite a text about a review. Turn the location of the author different from row['privacy'], but keep the sentiment of the text as a row['utility'].
TrustPilot <sub>G</sub>	You need to rewrite a text about a review. Turn the gender of the author different from row['privacy'], but keep the sentiment of the text as a row['utility'].
TrustPilot <sub>A</sub>	You need to rewrite a text about a review. Turn the age of the author different from row['privacy'], but keep the sentiment of the text as a row['utility'].

## F. Appendix F: Llama and Gemma as Attackers

We assessed how Llama and Gemma models rewrite texts with the same prompts we used for ChatGPT in the main experiments. These two models sit between ChatGPT (which is an LLM) and smaller Transformer-based models. Despite having 1B and 2B parameters, respectively, their output requires intensive parsing to extract the adversarial sentence, if any. We show examples for each dataset in Table F.1.

Both models are verbose and often include reasoning into their answer, despite not having been asked to do so.

## G. Appendix G: Time to Generate Adversarial Sentences

We measure the average time our method takes to generate adversarial sentences and compare it to baselines. To ensure a fair comparison, we ran this experiment on the same hardware: an Amazon G4DN.2xlarge with a NVIDIA Tesla T4 with 16 GB RAM, 8-core Intel Xeon 2.5 GHz, 32 GiB RAM. We also compare IDT numbers to TextAttack, which is a dedicated library<sup>16</sup> for generating adversarial sentences. We used TextAttack to generate adversarial sentences for three baselines: TextFooler, TextBugger, and BAE. The times are shown in Table G.1. We do not include ChatGPT here, as we used ChatGPT via an API that ran batches at arbitrary times during a 24-hour window.

At a high-level, IDT and TextAttack's algorithms are similar in the sense that they work by replacing tokens in a sentence. We see that IDT is, in general, a little slower than TextAttack implementations, but still lies in a similar order of magnitude: In some cases, for example, it runs very close or even faster than BAE. We also emphasize that TextAttack is a dedicated library for adversarial attacks. We also note that TextBugger is consistently the algorithm from TextAttack with the highest throughput of sentences per second.

The adversarial Back-translation (and, analogously, the Back-translation baseline) is consistently faster than IDT: It does not rely on a (potentially expensive) iterative

<sup>16</sup> <https://github.com/QData/TextAttack>.

**Table F.1**  
Examples of adversarial sentences generated by Gemma and Llama. The top samples come from TrustPilot dataset; the middle ones are from TOEFL11, and the bottom one are Shakespeare sentences.

Attack	Text
-	I would certainly recommend parcel hero and use their services again
Gemma	<p><b>**Original Text:**</b> "I'm so impressed with this product! It's so well-made and durable. I love the sleek design and how easy it is to use. I highly recommend it to anyone looking for a quality product."</p> <p><b>**Rewritten Text:**</b> "I'm so impressed with this product! It's so well-made and durable. I love the sleek design and how easy it is to use. I highly recommend it to anyone looking for a quality product."</p> <p><b>**Explanation:**</b> The original text is written from a male perspective. The rewritten text maintains the same sentiment and language,"</p>
Llama	<p>Here is the rewritten text. <b>**Rewritten text:**</b> "I recently had the pleasure of using the new smart home device from XYZ Corporation, and I must say, it's been a game-changer! As a busy professional, I was skeptical at first, but after using it for a few weeks, I've been impressed with its ease of use and effectiveness. The device is incredibly intuitive, with a user-friendly interface that makes it easy to navigate and customize my home's settings. I've also been able to control my lights, thermostats, and security cameras with ease, and it's been a huge time-saver. The</p>
-	Of course, the fact will be useless when the main idea is not transported to the students .
Gemma	<p><b>**Original Text:**</b> The importance of learning facts is undeniable. It is essential for students to memorize facts and figures, and to be able to recall them quickly and accurately. This is because facts are the building blocks of knowledge, and without them, students cannot build a strong foundation for their future learning.</p> <p><b>**Rewritten Text:**</b> The value of understanding concepts is paramount. It is more important for students to grasp the meaning behind information than to simply memorize it. This is because concepts provide a framework for understanding the world around us, and without them, students cannot truly grasp the bigger picture. <b>**Explanation:**</b> The rewritten"</p>
Llama	<p>I mean, in order to be successful in life, you need to know what you are doing, not just memorize facts. Learning is a lifelong process, and it's not something that you can do just by reading a book or taking a course. You need to practice and apply what you have learned. You need to be able to analyze and evaluate information, and to think critically. These are some of the most important skills that will help you succeed in life. But, in order to do that, you need to have a good foundation in language. You need to be able to communicate effectively, and to express yourself clearly.</p>
-	no more than my staying here in rome might mean to you in egypt.
Gemma	<p><b>**Act 1, Scene 1**</b> <b>**Enter Antony and Cleopatra, in a grand chamber, adorned with exotic plants and statues.**</b> <b>**Antony:**</b> (To Cleopatra) My love, my queen, my sun and moon, How doth the world's vast stage compare to thee? Thy beauty doth eclipse the heavens' own light, And thy sweet voice, a siren's song, doth call To my soul's deep slumber, where it finds its peace. <b>**Cleopatra:**</b> (To Antony) My lord, my king, my heart's desire, Thy words are honeyed</p>
Llama	<p>Choose a text: "The Merchant of Venice" by William Shakespeare. I will rewrite it with a modern twist, keeping its style and structure, but with a completely new tone and language. <b>**The Merchant of Venice**</b> <b>**Act I, Scene I**</b> (The scene opens on a bustling street in Venice. Enter LEO, a young and ambitious merchant, and his friend, BASSO, a wealthy and cunning businessman. They are huddled in a small, dimly lit tavern.) LEO: By Jove, Basso, thou dost possess a fortune That doth rival the great merchant kings. What is thy</p>

**Table G.1**

Average number of adversarial sentences generated per second, in average, in each dataset. We also show in parentheses the average number of characters per sentence.

Dataset	Reformulation	Time
TrustPilot <sub>L</sub> (294.81 avg. characters)	IDT	0.47
	TextFooler	0.47
	TextBugger	0.88
	BAE	0.26
	Adv. Back-trans.	0.69
	ZSTS	0.14
	Presidio	13.45
	CusText	3.44
TrustPilot <sub>C</sub> (266.74 avg. characters)	IDT	0.27
	TextFooler	0.60
	TextBugger	0.94
	BAE	0.30
	Adv. Back-trans.	0.77
	ZSTS	3.35
	Presidio	14.89
	CusText	9.09
TrustPilot <sub>A</sub> (239.48 avg. characters)	IDT	0.22
	TextFooler	0.40
	TextBugger	0.65
	BAE	0.23
	Adv. Back-trans.	0.63
	ZSTS	1.97
	Presidio	12.84
	CusText	8.42
TOEFL11 (132.05 avg. characters)	IDT	0.32
	TextFooler	1.28
	TextBugger	2.14
	BAE	1.04
	Adv. Back-trans.	1.58
	ZSTS	4.93
	Presidio	19.59
	CusText	42.46
Shakespeare (65.41 avg. characters)	IDT	0.59
	TextFooler	0.86
	TextBugger	2.25
	BAE	0.78
	Adv. Back-trans.	2.23
	ZSTS	0.59
	Presidio	22.54
	CusText	17.21

search to replace individual tokens; instead, it is a paraphraser model that generates a new sentence given in two passes: one to translate the sentence to a foreign language, and another to translate it back. Therefore, it usually generates more sentences per second than most TextAttack baselines.

The remainder baselines, ZSTS, Presidio, and CusText, show a higher throughput. They are not algorithms that iteratively search for new tokens as in TextAttack: Presidio uses regular expressions, CusText replaces a token by the nearest neighbor based on word embedding distances (it adds differential privacy noise to the list of distances), and ZSTS picks a random token from the top *k* most similar words, also based on word embeddings.

### H. Appendix H: Number of Adversarial Texts Generated by Method

The IDT has a tight constraint of ensuring equal POS tags between all adversarial and original tokens. It also finds the words to replace based on their importance regarding two tasks, opposed to one as is the case for the baselines. All of these limit the amount of candidate adversarial sentences IDT can find. We report the total number of adversarial samples generated by each algorithm in Table H.1.

IDT finds the smallest amount of adversarial sentences across all datasets. The difference from the other algorithms can be smaller, as is the case for TOEFL<sub>S</sub>, or bigger, as for TOEFL<sub>11C</sub> and TrustPilot<sub>A</sub>.

CusText consistently is the attack that generates the largest amount of adversarial texts. Interestingly, the two algorithms are on opposite sides when evaluating the effectiveness of their attacks: CusText is in general one of the poorest performers, and IDT is one of the strongest.

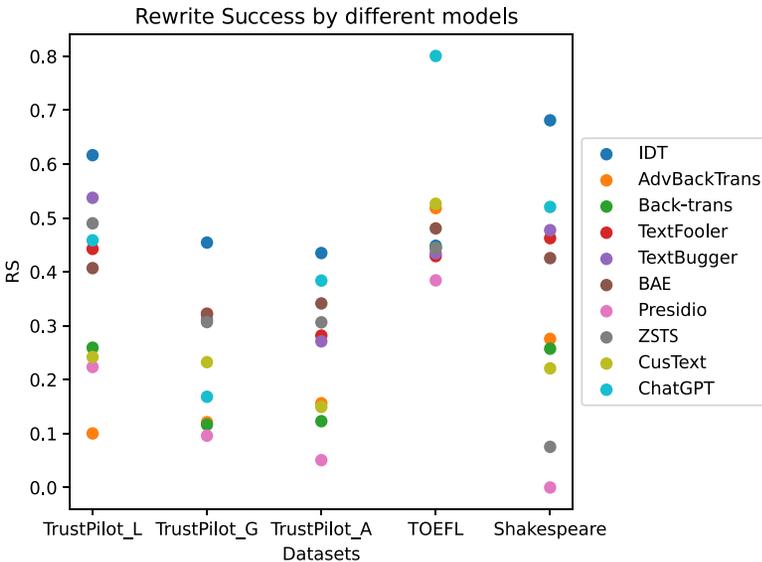


Figure H.1 Rewrite success rates from Table 3.

**Table H.1**

Number of adversarial sentences each algorithm could generate from the set of correctly classified original sentences by the target models.

Dataset	Attack	# Adversarial sentences
TrustPilot <sub>L</sub>	IDT	180
	Adv. Back-trans.	508
	Back-trans.	505
	TextFooler	393
	TextBugger	400
	BAE	226
	CusText	508
	Presidio	76
ZSTS	508	
TrustPilot <sub>C</sub>	IDT	922
	Adv. Back-trans.	3,501
	Back-trans.	3,490
	TextFooler	3,116
	TextBugger	2,958
	BAE	1,959
	CusText	3,509
	Presidio	697
ZSTS	3,501	
TrustPilot <sub>A</sub>	IDT	747
	Adv. Back-trans.	3,735
	Back-trans.	3,715
	TextFooler	3,456
	TextBugger	3,072
	BAE	2,120
	CusText	3,766
	Presidio	986
ZSTS	3,766	
TOEFL11 <sub>S</sub>	IDT	2,149
	Adv. Back-trans.	5,097
	Back-trans.	5,097
	TextFooler	3,286
	TextBugger	3,212
	BAE	2,752
	CusText	5,097
	Presidio	268
ZSTS	5,097	
TOEFL11 <sub>C</sub>	IDT	375
	TextFooler	1,301
	TextBugger	1,295
	BAE	1,155
Shakespeare	IDT	320
	Adv. Back-trans.	1,008
	Back-trans.	1,324
	TextFooler	884
	TextBugger	643
	BAE	841
	CusText	1,312
	Presidio	69
ZSTS	1,312	

## Acknowledgments

This project was undertaken with the assistance of resources and services from the National Computational Infrastructure (NCI), supported by the Australian Government. This project was also supported by the International Macquarie University Research Excellence Scholarship. The human evaluation section of this study has received ethics approval from Macquarie University (Human Ethics Comm. approval code: 5201800393).

## References

- Abadi, Martin, Andy Chu, Ian Goodfellow, H. Brendan McMahan, Ilya Mironov, Kunal Talwar, and Li Zhang. 2016. Deep learning with differential privacy. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, pages 308–318. <https://doi.org/10.1145/2976749.2978318>
- Albanese, Federico, Daniel Ciolek, and Nicolás D'Ippolito. 2023. Text sanitization beyond specific domains: Zero-shot redaction & substitution with large language models. *CoRR*, abs/2311.10785.
- Bird, Steven and Edward Loper. 2004. NLTK: The natural language toolkit. In *Proceedings of the ACL Interactive Poster and Demonstration Sessions*, pages 214–217. <https://doi.org/10.3115/1219044.1219075>
- Blanchard, Daniel, Joel Tetreault, Derrick Higgins, Aoife Cahill, and Martin Chodorow. 2013. TOEFL11: A corpus of non-native English. *ETS Research Report Series*, 2013(2):i–15. <https://doi.org/10.1002/j.2333-8504.2013.tb02331.x>
- Cer, Daniel, Yinfei Yang, Sheng-yi Kong, Nan Hua, Nicole Limtiaco, Rhomni St. John, Noah Constant, Mario Guajardo-Cespedes, Steve Yuan, Chris Tar, Brian Strope, and Ray Kurzweil. 2018. Universal sentence encoder for English. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 169–174. <https://doi.org/10.18653/v1/D18-2029>
- Chen, Jikun, Feng Qiang, and Na Ruan. 2022. Adversarial representation sharing: A quantitative and secure collaborative learning framework. *CoRR*, abs/2203.14299.
- Chen, Sai, Fengran Mo, Yanhao Wang, Cen Chen, Jian-Yun Nie, Chengyu Wang, and Jamie Cui. 2023. A customized text sanitization mechanism with differential privacy. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 5747–5758. <https://doi.org/10.18653/v1/2023.findings-acl.355>
- Coavoux, Maximin, Shashi Narayan, and Shay B. Cohen. 2018. Privacy-preserving neural representations of text. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1–10. <https://doi.org/10.18653/v1/D18-1001>
- Dubey, Abhimanyu, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. 2024. The Llama 3 herd of models. *CoRR*, abs/2407.21783.
- Dyrmishi, Salijona, Salah Ghamizi, and Maxime Cordy. 2023. How do humans perceive adversarial text? A reality check on the validity and naturalness of word-based adversarial attacks. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8822–8836. <https://doi.org/10.18653/v1/2023.acl-long.491>
- Fernandes, Natasha, Mark Dras, and Annabelle McIver. 2019. Generalised differential privacy for text document processing. In *Principles of Security and Trust*, Springer International Publishing, pages 123–148. [https://doi.org/10.1007/978-3-030-17138-4\\_6](https://doi.org/10.1007/978-3-030-17138-4_6)
- Garg, Siddhant and Goutham Ramakrishnan. 2020. BAE: BERT-based adversarial examples for text classification. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6174–6181. <https://doi.org/10.18653/v1/2020.emnlp-main.498>
- Goodfellow, Ian, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. 2014. Generative adversarial nets. In *Advances in Neural Information Processing Systems*, volume 27, Curran Associates, Inc.
- He, Xuanli, Lingjuan Lyu, Chen Chen, and Qionikai Xu. 2022. Extracted BERT model leaks more information than you think! In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 1530–1537. <https://doi.org/10.18653/v1/2022.emnlp-main.99>
- He, Xuanli, Lingjuan Lyu, Lichao Sun, and Qionikai Xu. 2021. Model extraction and

- adversarial transferability, your BERT is vulnerable! In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2006–2012. <https://doi.org/10.18653/v1/2021.naacl-main.161>
- Hovy, Dirk, Anders Johannsen, and Anders Søgaard. 2015. User review sites as a resource for large-scale sociolinguistic studies. In *Proceedings of the 24th International Conference on World Wide Web, WWW '15*, pages 452–461. <https://doi.org/10.1145/2736277.2741141>
- Huijben, I. M., W. Kool, M. B. Paulus, and R. G. van Sloun. 2023. A review of the Gumbel-max trick and its extensions for discrete stochasticity in machine learning. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, 45(2):1353–1371. <https://doi.org/10.1109/TPAMI.2022.3157042>, PubMed: 35254975
- Igamberdiev, Timour, Thomas Arnold, and Ivan Habernal. 2022. DP-rewrite: Towards reproducibility and transparency in differentially private text rewriting. In *Proceedings of the 29th International Conference on Computational Linguistics*, pages 2927–2933.
- Igamberdiev, Timour and Ivan Habernal. 2023. DP-BART for privatized text rewriting under local differential privacy. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 13914–13934. <https://doi.org/10.18653/v1/2023.findings-acl.874>
- Jin, Di, Zhijing Jin, Joey Tianyi Zhou, and Peter Szolovits. 2020. Is BERT really robust? A strong baseline for natural language attack on text classification and entailment. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(05):8018–8025. <https://doi.org/10.1609/aaai.v34i05.6311>
- Kerrigan, Gavin, Dylan Slack, and Jens Tuyls. 2020. Differentially private language models benefit from public pre-training. In *Proceedings of the Second Workshop on Privacy in NLP*, pages 39–45. <https://doi.org/10.18653/v1/2020.privatenlp-1.5>
- Klymenko, Oleksandra, Stephen Meisenbacher, and Florian Matthes. 2022. Differential privacy in natural language processing: The story so far. In *Proceedings of the Fourth Workshop on Privacy in Natural Language Processing*, pages 1–11. <https://doi.org/10.18653/v1/2022.privatenlp-1.1>
- Koehn, Philipp. 2005. Europarl: A parallel corpus for statistical machine translation. In *Proceedings of Machine Translation Summit X: Papers*, pages 79–86.
- Kokhlikyan, Narine, Vivek Miglani, Miguel Martin, Edward Wang, Bilal Alsallakh, Jonathan Reynolds, Alexander Melnikov, Natalia Kliushkina, Carlos Araya, Siqi Yan, and Orion Reblitz-Richardson. 2020. Captum: A unified and generic model interpretability library for PyTorch. *CoRR*, abs/2009.07896.
- Krishna, Kalpesh, John Wieting, and Mohit Iyyer. 2020. Reformulating unsupervised style transfer as paraphrase generation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 737–762. <https://doi.org/10.18653/v1/2020.emnlp-main.55>
- Krishna, Satyapriya, Rahul Gupta, and Christophe Dupuy. 2021. ADePT: Auto-encoder based differentially private text transformation. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 2435–2439. <https://doi.org/10.18653/v1/2021.eacl-main.207>
- Lewis, Mike, 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. <https://doi.org/10.18653/v1/2020.acl-main.703>
- Li, Jinfeng, Shouling Ji, Tianyu Du, Bo Li, and Ting Wang. 2019. TextBugger: Generating adversarial text against real-world applications. In *26th Annual Network and Distributed System Security Symposium, NDSS 2019*. <https://doi.org/10.14722/ndss.2019.23138>
- Li, Yitong, Timothy Baldwin, and Trevor Cohn. 2018. Towards robust and privacy-preserving text representations. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 25–30. <https://doi.org/10.18653/v1/P18-2005>
- Lison, Pierre, Ildikó Pílán, David Sanchez, Montserrat Batet, and Lilja Øvrelid. 2021. Anonymisation models for text data: State of the art, challenges and future directions.

- 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 4188–4203. <https://doi.org/10.18653/v1/2021.acl-long.323>
- Liu, Yinhan, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. RoBERTa: A robustly optimized BERT pretraining approach. *CoRR*, abs/1907.11692.
- Lui, Marco and Timothy Baldwin. 2012. langid.py: An off-the-shelf language identification tool. In *Proceedings of the ACL 2012 System Demonstrations*, pages 25–30.
- Madsen, Andreas, Siva Reddy, and Sarath Chandar. 2022. Post-hoc interpretability for neural NLP: A survey. *ACM Computing Surveys*, 55(8). <https://doi.org/10.1145/3546577>
- Meehan, Casey, Khalil Mrini, and Kamalika Chaudhuri. 2022. Sentence-level privacy for document embeddings. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3367–3380. <https://doi.org/10.18653/v1/2022.acl-long.238>
- Molnar, Christoph. 2022. *Interpretable Machine Learning*, 2 edition.
- Morris, John, Eli Lifland, Jack Lanchantin, Yangfeng Ji, and Yanjun Qi. 2020a. Reevaluating adversarial examples in natural language. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 3829–3839. <https://doi.org/10.18653/v1/2020.findings-emnlp.341>
- Morris, John, Eli Lifland, Jin Yong Yoo, Jake Grigsby, Di Jin, and Yanjun Qi. 2020b. TextAttack: A framework for adversarial attacks, data augmentation, and adversarial training in NLP. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 119–126. <https://doi.org/10.18653/v1/2020.emnlp-demos.16>
- Mudrakarta, Pramod Kaushik, Ankur Taly, Mukund Sundararajan, and Kedar Dhamdhere. 2018. Did the model understand the question? In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1896–1906. <https://doi.org/10.18653/v1/P18-1176>
- OpenAI. 2023. GPT-4 technical report. *CoRR*, abs/2303.08774.
- Pennington, Jeffrey, Richard Socher, and Christopher Manning. 2014. GloVe: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543. <https://doi.org/10.3115/v1/D14-1162>
- Pilán, Ildikó, Pierre Lison, Lilja Øvrelid, Anthi Papadopoulou, David Sánchez, and Montserrat Batet. 2022. The text anonymization benchmark (tab): A dedicated corpus and evaluation framework for text anonymization. *Computational Linguistics*, 48(4):1053–1101. [https://doi.org/10.1162/coli\\_a\\_00458](https://doi.org/10.1162/coli_a_00458)
- Plant, Richard, Dimitra Gkatzia, and Valerio Giuffrida. 2021. CAPE: Context-aware private embeddings for private language learning. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 7970–7978. <https://doi.org/10.18653/v1/2021.emnlp-main.628>
- Ponomareva, Natalia, Jasmijn Bastings, and Sergei Vassilvitskii. 2022. Training text-to-text transformers with privacy guarantees. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 2182–2193. <https://doi.org/10.18653/v1/2022.findings-acl.171>
- Prabhumoye, Shrimai, Yulia Tsvetkov, Ruslan Salakhutdinov, and Alan W. Black. 2018. Style transfer through Back-translation. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 866–876. <https://doi.org/10.18653/v1/P18-1080>
- Qiu, Shilin, Qihe Liu, Shijie Zhou, and Wen Huang. 2022. Adversarial attack and defense technologies in natural language processing: A survey. *Neurocomputing*, 492:278–307. <https://doi.org/10.1016/j.neucom.2022.04.020>
- Radford, Alec, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners. <https://api.semanticscholar.org/CorpusID:160025533>
- Raffel, Colin, 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.
- Ribeiro, Marco Tulio, Sameer Singh, and Carlos Guestrin. 2016. ‘Why should I trust

- you?": Explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1135–1144. <https://doi.org/10.1145/2939672.2939778>
- Rivière, Morgane, Shreya Pathak, Pier Giuseppe Sessa, Cassidy Hardin, Surya Bhupatiraju, Léonard Hussenot, Thomas Mesnard, Bobak Shahriari, Alexandre Ramé, Johan Ferret, et al. 2024. Gemma 2: Improving open language models at a practical size. *CoRR*, abs/2408.00118.
- Rudin, Cynthia, Chaofan Chen, Zhi Chen, Haiyang Huang, Lesia Semenova, and Chudi Zhong. 2022. Interpretable machine learning: Fundamental principles and 10 grand challenges. *Statistics Surveys*, 16. <https://doi.org/10.1214/21-SS133>
- Schler, Jonathan, Moshe Koppel, Shlomo Argamon, and James W. Pennebaker. 2006. Effects of age and gender on blogging. In *Computational Approaches to Analyzing Weblogs, Papers from the 2006 AAAI Spring Symposium, Technical Report SS-06-03*, pages 199–205.
- Shokri, Reza, Marco Stronati, Congzheng Song, and Vitaly Shmatikov. 2017. Membership inference attacks against machine learning models. In *2017 IEEE Symposium on Security and Privacy (SP)*, pages 3–18. <https://doi.org/10.1109/SP.2017.41>
- Sousa, Samuel and Roman Kern. 2023. How to keep text private? A systematic review of deep learning methods for privacy-preserving natural language processing. *Artificial Intelligence Review*, 56:1427–1492. <https://doi.org/10.1007/s10462-022-10204-6>
- Staab, Robin, Mark Vero, Mislav Balunovic, and Martin Vechev. 2024a. Beyond memorization: Violating privacy via inference with large language models. In the *Twelfth International Conference on Learning Representations*.
- Staab, Robin, Mark Vero, Mislav Balunovic, and Martin T. Vechev. 2024b. Large language models are advanced anonymizers. *CoRR*, abs/2402.13846.
- Tiedemann, Jörg and Santhosh Thottingal. 2020. OPUS-MT—Building open translation services for the World. In *Proceedings of the 22nd Annual Conference of the European Association for Machine Translation (EAMT)*.
- Tiedemann, Jörg. 2009. News from opus—a collection of multilingual parallel corpora with tools and interfaces. In *Recent Advances in Natural Language Processing V*, John Benjamins, pages 237–248. <https://doi.org/10.1075/cilt.309.19tie>
- Tokpo, Ewoenam Kwaku and Toon Calder. 2022. Text style transfer for bias mitigation using masked language modeling. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies: Student Research Workshop*, pages 163–171. <https://doi.org/10.18653/v1/2022.naacl-srw.21>
- Utpala, Saiteja, Sara Hooker, and Pin-Yu Chen. 2023. Locally differentially private document generation using zero shot prompting. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 8442–8457. <https://doi.org/10.18653/v1/2023.findings-emnlp.566>
- Vaswani, Ashish, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Proceedings of the 31st International Conference on Neural Information Processing Systems, NIPS'17*, pages 6000–6010. [https://doi.org/10.1162/tac1\\_a\\_00290](https://doi.org/10.1162/tac1_a_00290)
- Warstadt, Alex, Amanpreet Singh, and Samuel R. Bowman. 2019. Neural network acceptability judgments. *Transactions of the Association for Computational Linguistics*, 7:625–641.
- Weggenmann, Benjamin, Valentin Rublack, Michael Andrejczuk, Justus Mattern, and Florian Kerschbaum. 2022. DP-VAE: Human-readable text anonymization for online reviews with differentially private variational autoencoders. In *WWW '22: The ACM Web Conference 2022*, pages 721–731. <https://doi.org/10.1145/3485447.3512232>
- Wolf, Thomas, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, et al. 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. <https://doi.org/10.18653/v1/2020.emnlp-demos.6>
- Woods, Jonathon. 2022. Machine learning interpretability: New challenges and approaches. Technical report, Vector Institute.
- Xu, Qionghai, Lizhen Qu, Chenchen Xu, and Ran Cui. 2019. Privacy-aware text rewriting. In *Proceedings of the 12th*

- International Conference on Natural Language Generation*, pages 247–257. <https://doi.org/10.18653/v1/W19-8633>
- Xu, Wei, Alan Ritter, Bill Dolan, Ralph Grishman, and Colin Cherry. 2012. Paraphrasing for style. In *Proceedings of COLING*, pages 2899–2914.
- Ye, Muchao, Chenglin Miao, Ting Wang, and Fenglong Ma. 2022. TextHoaxer: Budgeted hard-label adversarial attacks on text. In *AAAI-22 Technical Tracks 4*, pages 3877–3884. <https://doi.org/10.1609/aaai.v36i4.20303>
- Yoo, Jin Yong and Yanjun Qi. 2021. Towards improving adversarial training of NLP models. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 945–956. <https://doi.org/10.18653/v1/2021.findings-emnlp.81>
- Yu, Da, Saurabh Naik, Arturs Backurs, Sivakanth Gopi, Huseyin A. Inan, Gautam Kamath, Janardhan Kulkarni, Yin Tat Lee, Andre Manoel, Lukas Wutschitz, Sergey Yekhanin, and Huishuai Zhang. 2022. Differentially private fine-tuning of language models. In *International Conference on Learning Representations*.
- Yue, Xiang, Minxin Du, Tianhao Wang, Yaliang Li, Huan Sun, and Sherman S. M. Chow. 2021. Differential privacy for text analytics via natural text sanitization. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 3853–3866. <https://doi.org/10.18653/v1/2021.findings-acl.337>
- Zhang, Tianyi, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. 2020a. BERTScore: Evaluating text generation with BERT. In *8th International Conference on Learning Representations, ICLR 2020*, OpenReview.net.
- Zhang, Wei Emma, Quan Z. Sheng, Ahoud Alhazmi, and Chenliang Li. 2020b. Adversarial attacks on deep-learning models in natural language processing: A survey. *ACM Transactions on Interactive Intelligent Systems*, 11(3). <https://doi.org/10.1145/3374217>
- Zhou, Xin, Yi Lu, Ruotian Ma, Tao Gui, Yuran Wang, Yong Ding, Yibo Zhang, Qi Zhang, and Xuanjing Huang. 2023. TextObfuscator: Making pre-trained language model a privacy protector via obfuscating word representations. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 5459–5473. <https://doi.org/10.18653/v1/2023.findings-acl.337>