Attacking Misinformation Detection Using Adversarial Examples Generated by Language Models

Piotr Przybyła[1,2] and **Euan McGill**[1] and **Horacio Saggion**[1]
[1] TALN Group, Universitat Pompeu Fabra, Barcelona, Spain
[2] Institute of Computer Science, Polish Academy of Sciences, Warsaw, Poland
{piotr.przybyla, euan.mcgill, horacio.saggion}@upf.edu

Abstract

Large language models have many beneficial applications, but can they also be used to attack content-filtering algorithms in social media platforms? We investigate the challenge of generating adversarial examples to test the robustness of text classification algorithms detecting low-credibility content, including propaganda, false claims, rumours and hyperpartisan news. We focus on simulation of content moderation by setting realistic limits on the number of queries an attacker is allowed to attempt. Within our solution (TREPAT), initial rephrasings are generated by large language models with prompts inspired by meaning-preserving NLP tasks, such as text simplification and style transfer. Subsequently, these modifications are decomposed into small changes, applied through beam search procedure, until the victim classifier changes its decision. We perform (1) quantitative evaluation using various prompts, models and query limits, (2) targeted manual assessment of the generated text and (3) qualitative linguistic analysis. The results confirm the superiority of our approach in the constrained scenario, especially in case of long input text (news articles), where exhaustive search is not feasible.

1 Introduction

Modern machine learning (ML) methods have proven effective in determining credibility of text in various scenarios (Horne and Adali, 2017; Graves, 2018; Al-Sarem et al., 2019; da San Martino et al., 2020; Barrón-Cedeño et al., 2024a), helping to tackle the challenge of misinformation (Lewandowsky et al., 2017; Tucker et al., 2018). Because of this development, many large platforms hosting user-generated data, e.g. social media, use text classifiers as part of their content moderation systems (Singhal et al., 2022). This raises the need to assess the *robustness* of such solutions, i.e. their ability to deliver correct result even for input manipulated by malicious actors. This is performed by seeking *adversarial examples* (AEs) – text samples modified in such a way that preserves their meaning, but elicits an incorrect response from the victim classifier (Carter et al., 2021).

A variety of experiments have been performed to confirm the vulnerability of credibility assessment to generic AE generation methods, and then seeking solutions tuned for this specific scenario within the *InCrediblAE* shared task (Przybyła et al., 2024c). The best approaches are based on iterative replacement of individual words with equivalents suggested by a language model. Generally, this direction has two major weaknesses.

Firstly, covering the vast space of possible rephrasing requires sending many queries to the victim system, sometimes several thousand, just to generate one adversarial AE. This makes the experiment stray far from a real-world implementation scenarios, where an adversary would be blocked from using the system when attempting to send so many queries. Secondly, word-replacement strategy can lead to poor meaning preservation. The manual evaluation of the shared task indicated that these methods often modify the meaning of the whole phrase, making such an AEs unusable.

However, AE generation is not the only task in Natural Language Processing (NLP) that requires modifying a given text while preserving its meaning, cf. text simplification (Shardlow, 2014), style transfer (Pang, 2019) or paraphrasing (Zhou and Bhat, 2021). The approaches using generative Large Language Models (LLMs) with carefully crafted prompts (Jayawardena and Yapa, 2024; Kew et al., 2023; Mukherjee et al., 2024) achieve the best results in these tasks.

Inspired by this work, here we propose TREPAT (**T**racing **RE**cursive **P**araphrasing for **A**dversarial examples from **T**ransformers): a solution for generating adversarial examples in English that leverages the LLMs' ability to reformulate a given text. The variants generated by LLMs are decomposed into

Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing, pages 27626–27642

atomic changes using Wagner-Fischer algorithm (Wagner and Fischer, 1974), which are then recursively applied using beam search (Lowerre, 1976), until the victim classifier changes its decision.

The contributions of this article are as follows:

1. A novel method for employing generative LLMs to obtain numerous variants of a given text fragment that could serve as AEs,
2. An investigation on which models and rephrasing prompts (focused on simplification, style change, paraphrasing etc.) return variants that change the victim's decision,
3. Manual annotation of the examples generated by various methods to verify their meaning preservation and language naturalness,
4. A linguistic analysis of the changes that LLMs perform in this scenario.

The code for TREPAT and annotation results are openly shared to encourage further research[1].

2 Related work

The investigation of AEs was initially proposed for image classification (Szegedy et al., 2013) and the extension of the framework to the text domain is challenging due to discrete nature of the medium (Zhang et al., 2020). In the misinformation domain, the fact-checking task was the first to be investigated for robustness (Hidey et al., 2020; Zhou et al., 2019), followed by fake news detection (Ali et al., 2021; Koenders et al., 2021).

A systematic analysis of AEs in credibility assessment, covering various tasks, attackers and victims, was performed through the BODEGA framework (Przybyła et al., 2024b), highlighting the vulnerabilities that affects also very large models. This line was extended through the *InCrediblAE* shared task (Przybyła et al., 2024c) organised at *CheckThat!* evaluation lab (Barrón-Cedeño et al., 2024b). The submitted solutions can be broadly divided into those relying on character changes (e.g. swapping *0* for *O*) (Valle Aguilera et al., 2024; Demirok et al., 2024; Guzman Piedrahita et al., 2024), replacing words according to the candidates from language models (He et al., 2024; Lewoniewski et al., 2024), or both (Roadhouse et al., 2024). We can also mention XARELLO (Przybyła et al., 2024a), which is using the BODEGA data, but with a different usage scenario: assuming that the attacker is performing multiple attacks on the same victim in the *adaptation* phase and thus can learn what modifications are successful. This process, powered by reinforcement learning, can allow to greatly reduce the number of queries in the test phase, but might not be possible if the victim is updated frequently. Additionally, the robustness analysis has also been performed to test attacks on the task of machine-generated text detection (Wang et al., 2024a).

Only one of the methods at InCrediblAE used an LLM model to generate rephrasings and the results were not satisfying (Demirok et al., 2024). Even beyond the misinformation detection, the abilities of LLMs have not yet been fully utilised for AE generation. We can mention their use to perform two subtasks: word importance ranking and synonym generation (Wang et al., 2024b). Moreover, *PromptAttack* (Xu et al., 2024) involves prompting an LLM to generate AEs, which makes it similar to our work. However, there is an important difference: PromptAttack assumes that the AEs are produced through interaction with the same model that is its victim. This approach cannot be applied to content filtering, where the model is inaccessible and might not even be a generative LLM.

Beyond the search for adversarial examples, there are other ways to 'attack' LLMs, e.g. looking for prompts that make them produce a desired output (e.g. toxic text) (Wallace et al., 2019; Zhu et al., 2024). However, there have been no successful approaches to use LLMs to generate reformulations that could be used as AEs to attack credibility assessment systems. This is the aim of TREPAT.

3 Methods

TREPAT explores adversarial examples in several simple steps (see Figure 1). Firstly, the text is split into smaller *fragments* (Section 3.1). Each of these fragments is then fed into an LLM with one of various prompts to obtain *rephrasings* (Section 3.2). Then, each rephrasing is decomposed into individual *changes*, which are then subsequently applied to the original text, resulting in *variants*, using the beam search procedure guided by the credibility score returned by the victim model (Section 3.3). Finally, if no AE has been found after testing all changes, the process is re-initiated with the best candidate seen so far as the starting point.

3.1 Splitting

Our preliminary tests have shown that LLMs, especially smaller ones, struggle to rephrase long

[1] https://github.com/piotrmp/trepat

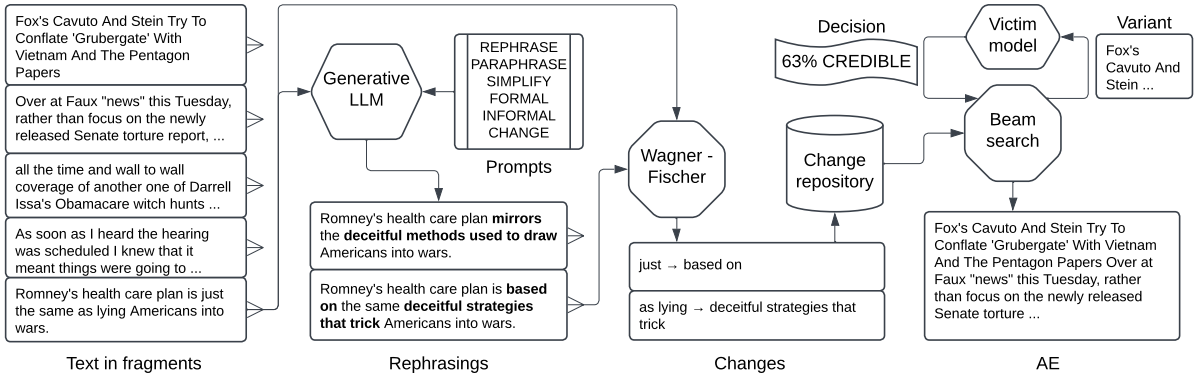


Figure 1: The architecture of TREPAT. A continuous text document is divided into fragments, each rephrased by a generative LLM (further processing only for one fragment is shown in the figure). The comparison of a rephrasing and text original text yields changes stored in a repository. These changes are iteratively applied following a beam search algorithm guided by the victim's response (i.e. binary credibility label).

sentences in a single round, often omitting important parts. Moreover, some of the input instances consist of whole documents containing multiple sentences, e.g. news articles (see section 4).

To divide input text into fragments fit for rephrasing, we perform the following splitting operations[2]:

1. Splitting the input types – this only applies to examples which combine the evidence and the claim in a single input (see section 4),
2. Splitting on newline characters,
3. Splitting into sentences with LAMBO (Przybyła, 2022),
4. Splitting on characters indicating phrase boundaries: dashes, quotation marks, commas and colons; as long as the results are at least 60 characters long.

For each of the fragments we also preserve its offset to be able to combine changes to different fragments in a single output text.

3.2 Rephrasing

The goal of this stage is to rephrase a given fragment in a way that changes its appearance but preserves the meaning. We use a LLM and provide it with a prompt corresponding to one of six commands, inspired with other text modification tasks:

- `REPHRASE`: the basic prompt, asking the model to *rephrase* the input fragment,
- `PARAPHRASE`: a variant aimed at stronger meaning preservation, asking for a *paraphrase*. LLMs have been shown to produce high-quality and diverse paraphrases (Jayawardena and Yapa, 2024).
- `SIMPLIFY`: a prompt requesting the model provide a simpler equivalent of the fragment. Previous work indicates that LLMs are able to handle this task based on a short prompt (with a few examples) (Kew et al., 2023).
- `FORMAL` and `INFORMAL`: variants requesting the model to rewrite the text in more (or less) formal style. Evaluations have shown that LLMs can achieve good-quality style transfer, at least in English (Mukherjee et al., 2024).
- `CHANGE`: a different phrasing, explicitly emphasising the need to *make changes* and relaxing the meaning preservation condition (*try to preserve* ...). In preliminary experiments this has led to more aggresive modifications.

The prompts were formulated through experimenting with GEMMA 1.0 2B model (Gemma Team and Google DeepMind, 2024). All of the prompts are quoted in Appendix A. Note how these prompts focus on the meaning-preservation goals, rather than expected modifications (e.g. *Replace at most two words in the sentence...*) used in PromptAttack (Xu et al., 2024).

We use six pre-trained instruction-tuned LLMs of various sizes, obtained through *HuggingFace Transformers* (Wolf et al., 2020):

[2]These rules were established based on a manual analysis of the output of the rephrasing module (for development portion of PR dataset, GEMMA-2B model, BiLSTM and BERT victims). We noticed that fragments shorter than 60 characters lacked the sufficient context for the LLM to rephrase, resulting in loss of meaning. Moreover, punctuation proved to be a good opportunity to separate longer phrases while maintaining semantic consistency.

- `LLAMA1B`: Llama 3.2[3] with 1 billion parameters (`meta-llama/Llama-3.2-1B-Instruct`)
- `GEMMA2B`: Gemma 2.0 (Gemma Team, 2024) with 2 billion parameters (`google/gemma-2-2b-it`),
- `LLAMA3B`: Llama 3.2 with 3 billion parameters (`meta-llama/Llama-3.2-3B-Instruct`),
- `OLMO7B`: OLMo (Groeneveld et al., 2024) v. 0724 with 7 billion parameters (`allenai/OLMo-7B-0724-Instruct-hf`),
- `LLAMA8B`: Llama 3.1 with 8 billion parameters (`meta-llama/Llama-3.1-8B-Instruct`),
- `GEMMA9B`: Gemma 2.0 with 9 billion parameters (`google/gemma-2-9b-it`),

The output of an LLM is parsed by splitting it into newline-separated reformulations and trimming unnecessary elements (enumerations, end-of-text tokens etc.).

3.3 Obtaining changes

Our preliminary experiments have shown that the reformulations generated by LLMs are usually not directly useful as AEs. They contain numerous modifications, while good-quality AEs can differ from the original example by only a single word. This is why we decompose the obtained reformulations into individual *changes*, each of which corresponds to a continuous sequence of tokens being replaced by a different sequence of tokens.

Take the following example:

- INPUT: *The recent rise of food prices is resulting in widespread discontent.*
- LLM OUTPUT: *The recent surge in food prices has caused widespread unease.*

This reformulation, performed by an LLM (GEMMA 2B), includes three changes[4]:

- *rise of -> surge in*
- *is resulting in -> has caused*
- *discontent -> unease*

The LLM has made multi-token changes, which would not be possible with methods based on word replacements, e.g. BERT-ATTACK.

In order to obtain these changes we convert both text fragments into sequences of tokens and then apply the Wagner-Fischer algorithm (Wagner and Fischer, 1974) for computing the edit distance. It represents a reformulation through the means of ADD, DELETE and REPLACE operations. We aggregate neighbouring operations to allow for multi-token changes, as shown above.

Finally, the changes are filtered by discarding those that contain only ADD or DELETE operations[5] or modify more than 2/3 of the fragment or 1/3 of the whole text.

3.4 Applying changes

The changes obtained from all reformulations of all fragments are collected in a single repository in order to be applied to the input text. Then, text *variants* are created by starting from the original text and gradually adding changes that modify it. Each created variant is sent as a query to the victim classifier and if it results in a modified response, it is returned as a successful AE.

In the example cited previously (*The recent rise of food prices is resulting in widespread discontent.*), the algorithm can check the victim's response to variants that include just one of the changes, e.g. *The recent **surge in** food prices is resulting in widespread discontent.* or *The recent rise of food prices **has caused** widespread discontent.* Then, variants combining two changes are possible, e.g. *The recent **surge in** food prices is resulting in widespread **unease**.*, with three changes, etc.

However, given the limitation of queries (see Section 4) and the number of possible changes in longer text examples, it is impossible to test all combinations to find the ones that change the victim's decision. Inspired by one of the solutions at the InCrediblAE shared task (Guzman Piedrahita et al., 2024), we apply beam search (Lowerre, 1976). This means that we record the *value* of each variant (i.e. the reduction of the probability of the original class according to victim classifier) and at any stage only k variants with the highest value are kept for applying further changes. The k is set to 5 to reduce the size of search scope[6].

A change is only applicable to a variant if the

[3] https://github.com/meta-llama/llama-models/blob/main/models/llama3_2/MODEL_CARD.md

[4] Note that changes can be context-sensitive, even if these examples appear general. Thus, in TREPAT we only apply changes to the sentences they were extracted from.

[5] These often correspond to reformulations where text content is moved within a fragment, which do not preserve meaning when decomposed into individual changes.

[6] This beam size corresponds to the typical number of atomic changes obtained from a single rephrasing, observed in the preliminary experiments. Thus, if only one useful rephrasing is generated, all of its changes can be explored.

part of the text it modifies has not been modified yet (by itself or another overlapping change). If at some point we run out of available changes, a new batch of reformulations is generated by the LLM from the variant of highest value so far.

4 Evaluation

The evaluation is performed using the BODEGA framework (Przybyła et al., 2024b), created to verify the robustness of credibility assessment solutions and based on previous corpora for English (Potthast et al., 2018; da San Martino et al., 2020; Thorne et al., 2018; Han et al., 2019). It covers four misinformation detection tasks: propaganda recognition (PR), fact-checking (FC), rumour detection (RD) and hyperpartisan news classification (HN). These tasks include text with various length and features: individual sentences (PR), claims with relevant evidence (FC), Twitter threads (RD) and news articles (HN). For each of these tasks, cast as binary classification, a model is trained using one of four popular architectures: BiLSTM neural network (Liu and Guo, 2019) and fine-tuned BERT (Devlin et al., 2018) or GEMMA (Gemma Team and Google DeepMind, 2024) in 2-billion and 7-billion variants.

BODEGA evaluates a given adversarial example by comparing it to the original text and measuring *confusion*, checking if the victim classifier changed its prediction; *semantic* similarity between the two texts using BLEURT (Sellam et al., 2020); and *character* similarity using Levenshtein distance (Levenshtein, 1966). All three scores are expressed as numbers in 0-1 range and are multiplied for a single BODEGA score, but can also be interpreted separately for better understanding of the results.

Additionally, we introduce a limit on the number of queries an attacker can perform to make our evaluation closer to real-world scenarios. The maximum numbers of posts that social media allow a user to submit are not disclosed, but estimated between 10 and 100 submissions per day[7]. We decided to generally allow 50 attempts, but also test how this parameter influences the performance (see Experiment 4).

4.1 Automatic experiments

We perform four experiments:

Experiment 1 aims to compare various LLMs (section 3.2) in the task. We use the REPHRASE prompt, and for each task take 400 examples from the development portion of the BODEGA datasets and compute BODEGA score averaged over all victim models. The results of this experiment guide the choice of an LLM for next steps.

Experiment 2 is designed to check which prompting strategy (section 3.2) is the most effective for generating AEs that achieve decision change and preserve the meaning. It also involves development data and follows the design of experiment 1, except we only test the LLM chosen there. The prompt (or prompts) selected based on these results will be used in final evaluation.

Experiment 3 plays the role of the main evaluation. It is based on the attack portion of the BODEGA datasets and compares TREPAT with parameters chosen as above with all the baselines. Unlike in the previous experiments, we analyse the results for each victim separately and include the partial scores – for confusion, semantic and character similarity.

Experiment 4 tests the applicability of the proposed method by checking the performance (BODEGA score averaged over victims and tasks) of TREPAT and baselines when different number of queries to a victim are allowed: 10, 50 (default used in experiments 1-3), 100 or 250.

4.2 Baselines

Based on the analysis of the previous work, the following solutions are used to compare to full TREPAT in experiments 3 and 4:

BERT-ATTACK (Li et al., 2020) was the overall best method in the original BODEGA evaluation, which covered a variety of AE generation approaches. It looks for replacements to a given word by applying language modelling through BERT.

F-BERT-ATTACK is our modification of the above to better fit the constrained query limit. The problematic aspect of BERT-ATTACK is its initial step, which selects the most vulnerable word by observing victim's response to its removal, requiring sending many queries for longer text. Here we replace this step by obtaining word importance randomly, allowing the attacker to perform viable attacks from the first query.

BeamAttack is a solution submitted by the *Text-Trojaners* team (Guzman Piedrahita et al., 2024) to the *InCrediblAE* shared task, obtaining nearly the highest score. Similarly to our solution, it employs

[7] E.g. https://help.simplified.com/en/articles/6067588-what-are-the-daily-posting-limits-on-each-social-media or https://support.buffer.com/article/646-daily-posting-limits

beam search to find the best replacement. We set its parameters to the lowest values considered by the authors to limit the search scope: 10 beams, 5 hypotheses and branching factor of 10.

TREPAT-simple is a simplified version of TREPAT, where LLM-generated rephrasings are used directly, instead of being split into changes and applied through beam search. The variant with the whole procedure is labelled **TREPAT-full**.

4.3 Manual evaluation

Evaluating meaning similarity is a difficult task and while automatic measures are being used in an equivalent role in machine translation, in InCrediblAE they were shown to poorly align with the human judgement in the adversarial example assessment. For this reason, we have decided to perform manual evaluation by asking human annotators to evaluate the quality of AEs generated by TREPAT, compared to other methods.

For that purpose, we take the output of Experiment 3 and randomly select 20% of the cases where successful AEs are available: from TREPAT and the best baseline method for this victim/task combination. We use two annotators, with random 25% of the data being assigned to both of them to measure agreement. Both our annotators are linguists: one a native speaker of English and the other one with certified proficiency in the language.

Each annotator is presented with a list of triples, consisting of the original text, modification A and modification B, where A and B are randomly taken from the baseline or TREPAT adversarial examples. The spans changed between the variants are highlighted. The annotators are then asked to decide which modification offers better **meaning preservation** (maintaining the meaning expressed in the original text) and language **naturalness** (seeming fluent, grammatical and authentic, as opposed to artificial and manipulated). While the latter criterion is well known in human evaluation of NLP solutions (Howcroft et al., 2020; Belz et al., 2020), the former is specific to AE assessment, but also found in evaluation of style transfer (Cao et al., 2020) and simplification (Stodden and Kallmeyer, 2022).

When the annotators are unable to choose between A and B, they can say that either 'Both' or 'Neither' of the options satisfies a given criterion. However, they are encouraged to make a clear decision even for small differences. Full annotation guidelines are included as Appendix C.

	BODEGA score			
LLM	**PR**	**FC**	**RD**	**HN**
LLAMA 1B	0.2297	0.3007	**0.1377**	**0.1691**
GEMMA 2B	0.2119	0.2316	0.0960	0.1512
LLAMA 3B	0.2231	**0.3062**	0.1188	0.1548
OLMO 7B	0.2420	0.3036	0.1313	0.1408
LLAMA 8B	0.2366	0.3038	0.1011	0.1584
GEMMA 9B	**0.2542**	0.3041	0.1285	0.1407

Table 1: Experiment 1 results, showing the BODEGA score of TREPAT with various LLMs, averaged over all victims trained for each task.

	BODEGA score			
Prompt	**PR**	**FC**	**RD**	**HN**
REPHRASE	0.2420	0.3035	0.1313	0.1420
PARAPHRASE	0.2361	0.3027	0.1221	0.1466
SIMPLIFY	0.2400	0.2909	0.1344	0.1567
FORMAL	0.2400	0.2939	**0.1493**	0.1525
INFORMAL	**0.2631**	**0.3242**	0.1298	**0.1780**
CHANGE	0.2478	0.3016	0.1286	0.1453
	Semantic score			
REPHRASE	0.7550	0.7830	0.8629	0.9443
PARAPHRASE	**0.7620**	0.7930	0.8534	0.9435
SIMPLIFY	0.7615	0.7899	**0.8687**	0.9431
FORMAL	0.7568	0.7953	0.8556	0.9350
INFORMAL	0.7508	**0.8032**	0.8611	0.9425
CHANGE	0.7618	0.7915	0.8676	**0.9456**

Table 2: Experiment 2 results, showing the BODEGA and semantic score of OLMO-based TREPAT with various prompts, averaged over victims for each task.

5 Results

The work performed offers us three perspectives for assessing the quality of the generated examples:

- automatic quantitative experiments, measuring the interaction between TREPAT and various attack victims (5.1),
- manual evaluation of the text quality, comparing our method with the strongest competitor, done blindly by proficient speakers (5.2),
- qualitative analysis of selected examples by a professional linguist to investigate the linguistic patterns present in the generations (5.3).

5.1 Automatic experiments

Experiment 1: Table 1 shows the result of the LLM selection. We can see that there is no one model that dominates across the board. Instead, in each task a different LLM achieves the best score and the differences between them are quite limited. Therefore, we have decided to use OLMO due to its open and transparent features (Groeneveld et al., 2024), as opposed to the commercial models.

Experiment 2: Table 2 includes the results, with the upper half showing the BODEGA score

Task	Prompt	BODEGA	Confusion	Semantic	Character	Queries
PR	BERT-ATTACK	**0.2307**	0.3462	0.7221	**0.9186**	40.4146
	F-BERT-ATTACK	0.2260	0.3462	0.7095	0.9154	36.1707
	BeamAttack	0.1711	0.2404	**0.7832**	0.8946	46.1220
	TREPAT-simple	0.1560	**0.5625**	0.5367	0.4620	25.8822
	TREPAT-full	**0.2307**	0.3870	0.7124	0.8159	27.9279
FC	BERT-ATTACK	0.2289	0.3086	0.7649	**0.9693**	46.6148
	F-BERT-ATTACK	0.1216	0.1679	0.7520	0.9622	45.4988
	BeamAttack	0.0876	0.1012	**0.8982**	0.9614	49.4938
	TREPAT-simple	**0.3783**	**0.6444**	0.7317	0.7785	25.3358
	TREPAT-full	0.3348	0.4444	0.8175	0.9167	33.5605
RD	BERT-ATTACK	0.0271	0.0530	0.5256	**0.9727**	48.9952
	F-BERT-ATTACK	0.0292	0.0627	0.4821	0.9705	47.9229
	BeamAttack	0.0308	0.0361	**0.8842**	0.9635	49.5942
	TREPAT-simple	0.0987	**0.1711**	0.6949	0.7450	44.5253
	TREPAT-full	**0.1176**	0.1422	0.8696	0.9411	45.3590
HN	BERT-ATTACK	0.0000	0.0000	0.0000	0.0000	50.0000
	F-BERT-ATTACK	0.0732	0.1100	0.6691	**0.9939**	46.2150
	BeamAttack	0.0000	0.0000	0.0000	0.0000	50.0000
	TREPAT-simple	**0.2646**	**0.3000**	0.9012	0.9772	38.1675
	TREPAT-full	0.1719	0.1850	**0.9362**	0.9920	44.1525

Table 3: Evaluation results showing the performance of TREPAT variants and baselines, applied to **BERT** victim models trained for the four tasks (results for BiLSTM, GEMMA2B and GEMMA7B victims are available in the appendix). For each run, the mean BODEGA, confusion, semantic and character scores are included, as well as the number of queries.

achieved with the TREPAT method using various prompt types. Interestingly, the best performance is achieved by prompts that perform style transfer towards a style that differs from the original text: INFORMAL rephrasing for text from journalistic (PR, HN) or encyclopaedic (FC) sources, and FORMAL for the task with social media messages (RD). Other approaches also perform well, but not quite as the style transfer.

Additionally, we verify whether the successful prompts do not harm the meaning preservation by checking the semantic score (bottom half of Table 2). They all seem quite similar in that respect, with the differences mostly within 1% range. For the final evaluation we choose the FORMAL (for RD) and INFORMAL (for PR, FC and HN) prompts.

Experiment 3: Table 3 shows the detailed attack summary for the BERT victim models. The results for the other victims (BiLSTM, GEMMA2B and GEMMA7B) are shown in Appendix D (Tables 5, 6 and 7) and paint a broadly similar picture.

We see that in propaganda recognition task, the simplest solution (BERT-ATTACK) works the best, or equally well as TREPAT-full in case of BERT victims. This task involves very short fragments (average length of 24.4 words), which means the 50 queries are sufficient to find an AE for a substantial number of cases (confusion score above 30%).

In case of fact-checking, the fragments are slightly longer (average of 41.3 words), since they include both a claim and evidence necessary to verify it. In this situation, both TREPAT variants achieve superior results. This involves finding AEs for more examples, even if not all of them have the highest semantic similarity. This aspect is better addressed by BeamSearch, but the lower number of successes (e.g. 10% in BERT-FC, compared to 64% of TREPAT-simple) limits the overall score.

The RD and HN tasks both include very long text fragments: Twitter threads (average of 320.4 words) and news articles (average of 708.6 words). BERT-ATTACK and BeamAttack are clearly constrained by the victim usage limit: the number of queries asked gets close to or reaches the limit of 50. For hyperpartisan news, this situation happens for every instance, resulting in a BODEGA score of 0.0. TREPAT-simple thrives in these conditions, delivering the best overall score in both tasks.

Taking into account all the victims (see Appendix D), the TREPAT variants obtain the highest BODEGA score for 15 out of 16 tested scenarios. The text length plays a role, with our approach dominating BERT-ATTACK for longer examples. The F-BERT-ATTACK variant has non-zero scores in more scenarios, but it achieves only one top spot. The BeamAttack's limited scope is reflected by confusion score, but where it is successful, the AEs have very high semantic similarity to the original text. The victims based on modern large LLMs are not necessarily more robust than BERT, aligning

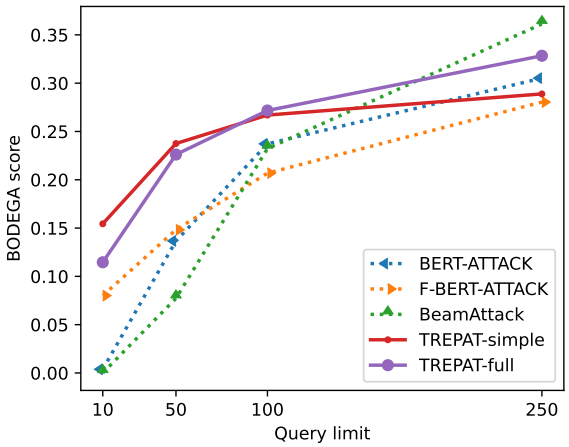


Figure 2: Experiment 4 results, showing the BODEGA score (averaged over victims and tasks) of various methods, evaluated with a given victim query limit (x axis).

with the observations for simpler AE generators in the same task (Przybyła et al., 2024b).

Regarding the TREPAT variants, we can see that each has its strengths. The simple version performs more aggressive rephrasing, reaching higher confusion rates with less queries, obtaining the best result for FC and HN. The full version gradually applies small changes, which requires more queries, but guarantees better semantic similarity, with the best BODEGA score for PR and RD.

Experiment 4: Figure 2 shows the performance of the tested methods for various limits of queries allowed for each example. We can see that TREPAT in both variants clearly outperforms baselines within 10-100 range reported as typical daily limits in social media sites. We need to allow 250 queries to see the advantage of methods designed for unlimited queries (BeamAttack).

5.2 Manual evaluation

The data prepared for manual evaluation according to the procedure in Section 4.3 included 350 instances: 165 from PR, 100 from FC, 41 from RD and 44 from HN. Of the 16 task/victim combinations, BERT-ATTACK was used as a baseline in 7, F-BERT-ATTACK in 8 and BeamAttack in 1.

We calculated the inter-annotator agreement by taking the cases where both annotators made a clear decision (A or B) and computing in how many of these they agree. The result is 75% for meaning preservation and 63% for language naturalness, reflecting the subjective nature of the task.

Table 4 shows the results of the manual annotation, computed based on cases where either one or both annotators preferred one of the options. We can see that in all tasks, the annotators judged the changes proposed by TREPAT as better at preserving meaning. With the exception of fact-checking, TREPAT also offered more natural language, confirming the validity of the attacks.

	PR	FC	RD	HN
	Meaning preservation			
Baseline	36.43%	34.18%	30.56%	25.00%
TREPAT	**63.57%**	**65.82%**	**69.44%**	**75.00%**
	Language naturalness			
Baseline	47.48%	**59.52%**	48.48%	40.63%
TREPAT	**52.52%**	40.48%	**51.52%**	**59.38%**

Table 4: Results of the manual evaluation in each task, expressed as a percentage of cases where either TREPAT or the baseline approach were chosen as preferred.

It is interesting to notice that in terms of meaning preservation, the proposed method has the biggest advantage over the baseline in the rumour and hyperpartisan news detection tasks. These are also the tasks that are the most challenging according to the automatic evaluation (see Section 5.1), with the low confusion score of both TREPAT and the baselines expressing the difficulty of finding an AE within the limited number of queries.

In terms of language naturalness, the gains are not as strong and in some cases the baselines offer more believable language, especially for fact-checked claims. See further discussion on the failures observed in Section 5.3.

5.3 Linguistic analysis

In order to be robust in real-world scenario, AEs must also produce utterances which are grammatical and authentic and also believable according to the consumer's world knowledge. TREPAT was the superior approach in terms of meaning preservation and was favoured most of the time for naturalness, using many strategies to modify the base text.

In this section, we describe frequent strategies that the attackers use to adhere to the heuristics set out for manual evaluation along with exemplars which can be found in Appendix B. Tokens and characters which have been changed between the original and modified texts are in **bold** text.

For an example of a successful modification, consider example (1) in Appendix B from the PR task: we observe that TREPAT uses a familiar rephrasing "rowed back"→"it revised" and semantically bleaches the noun phrase "verified facts" to "confirmed data". This does not alter the text meaning.

Compare it to BERT-ATTACK, which changes only one word, the proper noun "Guardian" to "forward". Unlike the TREPAT rephrasing, this seemingly light-touch approach is unsuccessful and jarring to a reader by not only making the phrase ungrammatical - but also removing a key piece of information from the phrase (a newspaper name). This strategy has been observed in other AE studies (Przybyła et al., 2024a) for this model.

TREPAT rephrasing is based on LLMs prone to hallucination, however. Examples (2a) and (2b) from FC shows that this can affect naturalness, with a noun phrase which appears misplaced or unnecessarily repeated. For example, TREPAT often introduces repetition of information (Example (3)) or individual words – such as "*their sincerely-held **belief** beliefs" from HN – in its paraphrasing. As shown in Table 4, TREPAT retains meaning better but is considered less natural for the FC task.

Two of the modification tasks described in Section 3.2 – introducing a more formal or informal style – are mostly successful (Example (4)), especially in retaining meaning and naturalness with a couple of exceptions (Example (5)).

TREPAT AEs appear to be less liable to violate grammatical rules like verbal agreement compared to other models. For example, BERT-ATTACK in Example (4) "*they know**s**" or F-BERT-ATTACK in the RD task "*make them suffer**ing** a trial". Many generate highly idiomatic structures and phrases such as "right in the gut" in the PR task.

6 Conclusions

We have presented a method to harness the text generation potential of large language model and apply it to the task of generating adversarial examples. It can to attack misinformation detection classifiers, while maintaining realistic limits on the interaction with the victim and preserving the meaning of the original example. While the use of LLM for adversarial examples has been explored before, it has not been successful (Demirok et al., 2024) . Our work is the first to show that this method can improve the results, establishing SOTA on the BODEGA task. We are also the first to propose splitting rephrasings into edit operations to preserve semantic content. Interestingly, the large modern models are not necessarily more robust to our attacks, emphasising the need to analyse the vulnerability of any ML solutions deployed in such a sensitive role as content moderation in social media.

Acknowledgements

The work of P. Przybyła is part of the ERINIA project, which received funding from the European Union's Horizon Europe research and innovation programme under grant agreement No 101060930. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the funders. Neither the European Union nor the granting authority can be held responsible for them. We gratefully acknowledge Polish high-performance computing infrastructure PLGrid (HPC Centers: ACK Cyfronet AGH) for providing computer facilities and support within computational grant no. PLG/2025/018019. We also acknowledge support from Departament de Recerca i Universitats de la Generalitat de Catalunya (ajuts SGR-Cat 2021) and from Maria de Maeztu Units of Excellence Programme CEX2021-001195-M, funded by MCIN/AEI /10.13039/501100011033. Finally, we are grateful for the participation of Alba Táboas García in the manual evaluation effort.

Ethical impact

As with any research in the area of credibility and misinformation, we need to consider whether our work can be useful for malicious actors. We do try to make our attack scenario as close to real world as possible (e.g. limiting the number of queries), but several differences remain that make it impossible to use our method directly for performing attacks. The chief among them is the usage of victim's decision as a continuous number (e.g. 27% credible) instead of a binary decision (e.g. REJECTED) that would typically be the only output seen by a user. This is a conscious decision, resulting from trying to balance the realistic setup with avoiding creating tools that can be directly used by attackers. Requiring a continuous score lets us provide a solution that can be used by the services deploying content filtering models, but not the attackers. This assumption is common in AE generation field, including the framework we use for evaluation.

Moreover, we need to note that the very practice of using automatic ML-based solutions for content filtering is considered unethical by many, e.g. because of equivalence with censorship according to the international law (Llansó, 2020). Nevertheless, it remains widely used by platforms and may be unavoidable given the amount of content they need to scrutinise.

Limitations

The results show that TREPAT works as expected, delivering many adversarial examples in limited query scenarios, even when dealing with very long text. However, some limitations remain.

Firstly, while we have modified BERT-ATTACK to make it a better fit for the constrained queries scenario, no equally obvious modification was performed for BeamAttack. Employing the reduced parameter setting was clearly not enough and we see the method attempting to send too many queries. We expect this and other methods could be tuned to deliver better AEs in this setting, but this is left for future work.

Moreover, a manual analysis of the results points to an important limitation of LLMs compared to simpler models: they often avoid generating text that might be considered sensitive, toxic or crude, resorting to euphemistic replacements. Unfortunately, such topics are prevalent in discussions adjacent to misinformation. The indirect paraphrases proposed by LLMs do not fit the context style and stray too far from the original to be useful for the task. Most of these are then removed through the filtering mechanism in TREPAT (section 3.3), but future work may lead to better solutions. This relates to a wider topic of LLM *exaggerated safety* (Chehbouni et al., 2024).

It is important to note that while here we argue that the limited-query scenario is close to the real-world situation, other attack scenarios may be applicable. For example, one in which an attacker uses a history of previous attacks on the same victim to be able to deliver more precise AEs later. Situations when unlimited queries are allowed can also happen, e.g. if a filtering system is open enough for an attacker to deploy a local copy for their use. This only emphasises that every time a text classifier is used in adversarial scenario, it needs to be first tested for robustness against attacks that are possible in this particular application.

Finally, we need to note that while the simulated nature of the AE search performed here and in other works in the domain remains its strong limitation, there is no clear alternative. One might imagine an attempt to attack real-world systems in order to test their robustness against certain prohibited content types (e.g. misinformation), but that could cause significant harm. Such actions would be illegal in most countries, breaking the terms and conditions of the services, and risking introducing further misinformation into the media, if successful. Therefore, attempting to simulate attacks under largely realistic conditions remains the best way to make services robust against adversarial actors.

References

Mohammed Al-Sarem, Wadii Boulila, Muna Al-Harby, Junaid Qadir, and Abdullah Alsaeedi. 2019. Deep learning-based rumor detection on microblogging platforms: A systematic review. *IEEE Access*, 7:152788–152812.

Hassan Ali, Muhammad Suleman Khan, Amer AlGhadhban, Meshari Alazmi, Ahmad Alzamil, Khaled Alutaibi, and Junaid Qadir. 2021. All Your Fake Detector are Belong to Us: Evaluating Adversarial Robustness of Fake-News Detectors Under Black-Box Settings. *IEEE Access*, 9:81678–81692.

Alberto Barrón-Cedeño, Firoj Alam, Tanmoy Chakraborty, Tamer Elsayed, Preslav Nakov, Piotr Przybyła, Julia Maria Struß, Fatima Haouari, Maram Hasanain, Federico Ruggeri, Xingyi Song, and Reem Suwaileh. 2024a. The CLEF-2024 CheckThat! Lab: Check-Worthiness, Subjectivity, Persuasion, Roles, Authorities, and Adversarial Robustness. In *Proceedings of the 46th European Conference on Information Retrieval (ECIR 2024)*, pages 449–458, Glasgow, UK. Springer Cham.

Alberto Barrón-Cedeño, Firoj Alam, Julia Maria Struß, Preslav Nakov, Tanmoy Chakraborty, Tamer Elsayed, Piotr Przybyła, Tommaso Caselli, Giovanni Da San Martino, Fatima Haouari, Chengkai Li, Jakub Piskorski, Federico Ruggeri, Xingyi Song, and Reem Suwaileh. 2024b. Overview of the CLEF-2024 CheckThat! Lab: Check-Worthiness, Subjectivity, Persuasion, Roles, Authorities and Adversarial Robustness. In *Experimental IR Meets Multilinguality, Multimodality, and Interaction. Proceedings of the Fifteenth International Conference of the CLEF Association (CLEF 2024)*.

Anya Belz, Simon Mille, and David M Howcroft. 2020. Disentangling the Properties of Human Evaluation Methods: A Classification System to Support Comparability, Meta-Evaluation and Reproducibility Testing. In *Proceedings of the 13th International Conference on Natural Language Generation*, pages 183–194, Dublin, Ireland. Association for Computational Linguistics.

Yixin Cao, Ruihao Shui, Liangming Pan, Min-Yen Kan, Zhiyuan Liu, and Tat-Seng Chua. 2020. Expertise Style Transfer: A New Task Towards Better Communication between Experts and Laymen. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1061–1071, Online. Association for Computational Linguistics.

Matthew Carter, Michail Tsikerdekis, and Sherali Zeadally. 2021. Approaches for Fake Content De-

tection: Strengths and Weaknesses to Adversarial Attacks. *IEEE Internet Computing*, 25(2):73–83.

Khaoula Chehbouni, Megha Roshan, Emmanuel Ma, Futian Wei, Afaf Taik, Jackie Cheung, and Golnoosh Farnadi. 2024. From Representational Harms to Quality-of-Service Harms: A Case Study on Llama 2 Safety Safeguards. In *Findings of the Association for Computational Linguistics ACL 2024*, pages 15694–15710, Bangkok, Thailand and virtual meeting. Association for Computational Linguistics.

Giovanni da San Martino, Alberto Barrón-Cedeño, Henning Wachsmuth, Rostislav Petrov, and Preslav Nakov. 2020. SemEval-2020 Task 11: Detection of Propaganda Techniques in News Articles. In *Proceedings of the Fourteenth Workshop on Semantic Evaluation (SemEval-2020)*, pages 1377–1414.

Basak Demirok, Mucahid Kutlu, Selin Mergen, and Bugra Oz. 2024. TurQUaz at CheckThat! 2024: Creating Adversarial Examples using Genetic Algorithm. In *Working Notes of CLEF 2024 - Conference and Labs of the Evaluation Forum*, CLEF˜2024, Grenoble, France.

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

Gemma Team. 2024. Gemma. DOI: 10.34740/KAGGLE/M/3301.

Gemma Team and Google DeepMind. 2024. Gemma: Open Models Based on Gemini Research and Technology. Technical report, Google DeepMind.

Lucas Graves. 2018. Understanding the Promise and Limits of Automated Fact-Checking. Technical report, Reuters Institute, University of Oxford.

Dirk Groeneveld, Iz Beltagy, Evan Walsh, Akshita Bhagia, Rodney Kinney, Oyvind Tafjord, Ananya Jha, Hamish Ivison, Ian Magnusson, Yizhong Wang, Shane Arora, David Atkinson, Russell Authur, Khyathi Chandu, Arman Cohan, Jennifer Dumas, Yanai Elazar, Yuling Gu, Jack Hessel, Tushar Khot, William Merrill, Jacob Morrison, Niklas Muennighoff, Aakanksha Naik, Crystal Nam, Matthew Peters, Valentina Pyatkin, Abhilasha Ravichander, Dustin Schwenk, Saurabh Shah, William Smith, Emma Strubell, Nishant Subramani, Mitchell Wortsman, Pradeep Dasigi, Nathan Lambert, Kyle Richardson, Luke Zettlemoyer, Jesse Dodge, Kyle Lo, Luca Soldaini, Noah Smith, and Hannaneh Hajishirzi. 2024. OLMo: Accelerating the Science of Language Models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 15789–15809, Bangkok, Thailand. Association for Computational Linguistics.

David Guzman Piedrahita, Arnisa Fazla, and Lucas Krauter. 2024. TextTrojaners at CheckThat! 2024: Robustness of Credibility Assessment with Adversarial Examples through BeamAttack. In *Working Notes of CLEF 2024 - Conference and Labs of the Evaluation Forum*, CLEF 2024, Grenoble, France.

Sooji Han, Jie Gao, and Fabio Ciravegna. 2019. Neural language model based training data augmentation for weakly supervised early rumor detection. In *Proceedings of the 2019 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining, ASONAM 2019*, pages 105–112. Association for Computing Machinery, Inc.

Haokun He, Yafeng Song, and Dylan Massey. 2024. Palöri at CheckThat! 2024 Shared Task 6: GloTa - Combining GloVe Embeddings with RoBERTa For Adversarial Attack. In *Working Notes of CLEF 2024 - Conference and Labs of the Evaluation Forum*, CLEF˜2024, Grenoble, France.

Christopher Hidey, Tuhin Chakrabarty, Tariq Alhindi, Siddharth Varia, Kriste Krstovski, Mona Diab, and Smaranda Muresan. 2020. DeSePtion: Dual Sequence Prediction and Adversarial Examples for Improved Fact-Checking. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8593–8606. Association for Computational Linguistics (ACL).

Benjamin D. Horne and Sibel Adali. 2017. This Just In: Fake News Packs a Lot in Title, Uses Simpler, Repetitive Content in Text Body, More Similar to Satire than Real News. In *Proceedings of the 2nd International Workshop on News and Public Opinion at ICWSM*. Association for the Advancement of Artificial Intelligence.

David M Howcroft, Anya Belz, Miruna-Adriana Clinciu, Dimitra Gkatzia, Sadid A Hasan, Saad Mahamood, Simon Mille, Emiel van Miltenburg, Sashank Santhanam, and Verena Rieser. 2020. Twenty Years of Confusion in Human Evaluation: NLG Needs Evaluation Sheets and Standardised Definitions. In *Proceedings of the 13th International Conference on Natural Language Generation*, pages 169–182, Dublin, Ireland. Association for Computational Linguistics.

Lasal Jayawardena and Prasan Yapa. 2024. Parafusion: A Large-Scale LLM-Driven English Paraphrase Dataset Infused with High-Quality Lexical and Syntactic Diversity. In *Artificial Intelligence and Big Data*, AIBD, pages 219–238. Academy & Industry Research Collaboration Center.

Tannon Kew, Alison Chi, Laura Vásquez-Rodríguez, Sweta Agrawal, Dennis Aumiller, Fernando Alva-Manchego, and Matthew Shardlow. 2023. BLESS: Benchmarking Large Language Models on Sentence Simplification. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*.

Camille Koenders, Johannes Filla, Nicolai Schneider, and Vinicius Woloszyn. 2021. How Vulnerable Are Automatic Fake News Detection Methods to Adversarial Attacks? *arXiv:2107.07970*.

Vladimir Iosifovich Levenshtein. 1966. Binary codes capable of correcting deletions, insertions, and reversals. *Soviet Physics Doklady*, 10:707–710.

Stephan Lewandowsky, Ullrich K.H. Ecker, and John Cook. 2017. Beyond Misinformation: Understanding and Coping with the "Post-Truth" Era. *Journal of Applied Research in Memory and Cognition*, 6(4):353–369.

Włodzimierz Lewoniewski, Piotr Stolarski, Milena Stróżyna, Elżbieta Lewańska, Aleksandra Wojewoda, Ewelina Księżniak, and Marcin Sawiński. 2024. OpenFact at CheckThat! 2024: Combining Multiple Attack Methods for Effective Adversarial Text Generation. In *Working Notes of CLEF 2024 - Conference and Labs of the Evaluation Forum*, CLEF˜2024, Grenoble, France.

Linyang Li, Ruotian Ma, Qipeng Guo, Xiangyang Xue, and Xipeng Qiu. 2020. BERT-ATTACK: Adversarial Attack Against BERT Using BERT. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6193–6202. Association for Computational Linguistics.

Gang Liu and Jiabao Guo. 2019. Bidirectional LSTM with attention mechanism and convolutional layer for text classification. *Neurocomputing*, 337:325–338.

Emma J Llansó. 2020. No amount of "AI" in content moderation will solve filtering's prior-restraint problem. *Big Data and Society*, 7(1).

Bruce T Lowerre. 1976. *The Harpy Speech Recognition System*. Ph.D. thesis, Carnegie Mellon University.

Sourabrata Mukherjee, Atul Kr. Ojha, and Ondrej Dusek. 2024. Are Large Language Models Actually Good at Text Style Transfer? In *Proceedings of the 17th International Natural Language Generation Conference*, pages 523–539, Tokyo, Japan. Association for Computational Linguistics.

Richard Yuanzhe Pang. 2019. The Daunting Task of Real-World Textual Style Transfer Auto-Evaluation. In *EMNLP Workshop on Neural Generation and Translation (WNGT 2019)*.

Martin Potthast, Johannes Kiesel, Kevin Reinartz, Janek Bevendorff, and Benno Stein. 2018. A Stylometric Inquiry into Hyperpartisan and Fake News. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 231–240. Association for Computational Linguistics.

Piotr Przybyła. 2022. LAMBO: Layered Approach to Multi-level BOundary identification. https://gitlab.clarin-pl.eu/syntactic-tools/lambo.

Piotr Przybyła, Euan McGill, and Horacio Saggion. 2024a. Know Thine Enemy: Adaptive Attacks on Misinformation Detection Using Reinforcement Learning. In *Proceedings of the 14th Workshop on Computational Approaches to Subjectivity, Sentiment, & Social Media Analysis*, pages 125–140, Bangkok, Thailand. Association for Computational Linguistics.

Piotr Przybyła, Alexander Shvets, and Horacio Saggion. 2024b. Verifying the robustness of automatic credibility assessment. *Natural Language Processing*, pages 1–29.

Piotr Przybyła, Ben Wu, Alexander Shvets, Yida Mu, Kim Cheng Sheang, Xingyi Song, and Horacio Saggion. 2024c. Overview of the CLEF-2024 CheckThat! Lab Task 6 on Robustness of Credibility Assessment with Adversarial Examples (InCrediblAE). In *Working Notes of CLEF 2024 - Conference and Labs of the Evaluation Forum*, CLEF˜2024, Grenoble, France.

Charlie Roadhouse, Matthew Shardlow, and Ashley Williams. 2024. MMU NLP at CheckThat! 2024: Homoglyphs are Adversarial Attacks. In *Working Notes of CLEF 2024 - Conference and Labs of the Evaluation Forum*, CLEF˜2024, Grenoble, France.

Thibault Sellam, Dipanjan Das, and Ankur Parikh. 2020. BLEURT: Learning Robust Metrics for Text Generation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7881–7892, Online. Association for Computational Linguistics.

Matthew Shardlow. 2014. A Survey of Automated Text Simplification. *International Journal of Advanced Computer Science and Applications*, 4(1).

Mohit Singhal, Chen Ling, Pujan Paudel, Poojitha Thota, Nihal Kumarswamy, Gianluca Stringhini, and Shirin Nilizadeh. 2022. SoK: Content Moderation in Social Media, from Guidelines to Enforcement, and Research to Practice. In *The 8th IEEE European Symposium on Security and Privacy (EuroS&P 2023)*. IEEE.

Regina Stodden and Laura Kallmeyer. 2022. TS-ANNO: An Annotation Tool to Build, Annotate and Evaluate Text Simplification Corpora. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 145–155, Dublin, Ireland. Association for Computational Linguistics.

Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. 2013. Intriguing properties of neural networks. *arXiv: 1312.6199*.

James Thorne, Andreas Vlachos, Oana Cocarascu, Christos Christodoulopoulos, and Arpit Mittal. 2018. The Fact Extraction and VERification (FEVER) Shared Task. In *Proceedings of the First Workshop on Fact Extraction and VERification (FEVER)*.

Joshua A. Tucker, Andrew Guess, Pablo Barberá, Cristian Vaccari, Alexandra Siegel, Sergey Sanovich, Denis Stukal, and Brendan Nyhan. 2018. Social Media, Political Polarization, and Political Disinformation: A Review of the Scientific Literature. Technical report, Hewlett Foundation.

José Valle Aguilera, Alberto J Gutiérrez Megías, Salud María Jiménez Zafra, Luis Alfonso Ureña López, and Eugenio Martínez Cámara. 2024. SINAI at CheckThat! 2024: Stealthy Character-Level Adversarial Attacks Using Homoglyphs and Search, Iterative. In *Working Notes of CLEF 2024 - Conference and Labs of the Evaluation Forum*, CLEF˜2024, Grenoble, France.

Robert A Wagner and Michael J Fischer. 1974. The String-to-String Correction Problem. *Journal of the ACM*, 21(1):168–173.

Eric Wallace, Shi Feng, Nikhil Kandpal, Matt Gardner, and Sameer Singh. 2019. Universal Adversarial Triggers for Attacking and Analyzing NLP. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 2153–2162, Hong Kong, China. Association for Computational Linguistics.

James Liyuan Wang, Ran Li, Junfeng Yang, and Chengzhi Mao. 2024a. RAFT: Realistic Attacks to Fool Text Detectors. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 16923–16936, Miami, Florida, USA. Association for Computational Linguistics.

Zimu Wang, Wei Wang, Qi Chen, Qiufeng Wang, and Anh Nguyen. 2024b. Generating Valid and Natural Adversarial Examples with Large Language Models. In *27th International Conference on Computer Supported Cooperative Work in Design (CSCWD)*, pages 1716–1721.

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

Xilie Xu, Keyi Kong, Ning Liu, Lizhen Cui, Di Wang, Jingfeng Zhang, and Mohan S Kankanhalli. 2024. An LLM can Fool Itself: A Prompt-Based Adversarial Attack. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net.

Wei Emma Zhang, Quan Z. Sheng, Ahoud Alhazmi, and Chenliang Li. 2020. Adversarial Attacks on Deep-learning Models in Natural Language Processing. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 11(3).

Jianing Zhou and Suma Bhat. 2021. Paraphrase Generation: A Survey of the State of the Art. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 5075–5086, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Zhixuan Zhou, Huankang Guan, Meghana Moorthy Bhat, and Justin Hsu. 2019. Fake News Detection via NLP is Vulnerable to Adversarial Attacks. In *ICAART 2019 - Proceedings of the 11th International Conference on Agents and Artificial Intelligence*, volume 2, pages 794–800. SciTePress.

Kaijie Zhu, Jindong Wang, Jiaheng Zhou, Zichen Wang, Hao Chen, Yidong Wang, Linyi Yang, Wei Ye, Yue Zhang, Neil Gong, and Xing Xie. 2024. PromptRobust: Towards Evaluating the Robustness of Large Language Models on Adversarial Prompts. In *Proceedings of the 1st ACM Workshop on Large AI Systems and Models with Privacy and Safety Analysis*, LAMPS '24, pages 57–68, New York, NY, USA. Association for Computing Machinery.

A Prompts

The following prompts were used to obtain rephrasings of a given fragment:

- REPHRASE: ***Rephrase*** *the provided input text. You can add, remove or replace individual words or punctuation characters, but* ***keep the changes to the minimum to preserve the original meaning****. Return five different rephrasings, separated by newline. Do not generate any text except the reformulations.*
  INPUT:
  <fragment>
  OUTPUT:
- PARAPHRASE: ***Paraphrase*** *the provided input text. You can add, . . .* (continued as above)
- SIMPLIFY: ***Simplify*** *the provided input text. You can add, . . .*
- FORMAL: *Rewrite the provided input text in a* ***more formal style****. You can add, . . .*
- INFORMAL: *Rewrite the provided input text in a* ***less formal style****. You can add, . . .*
- CHANGE: ***Make changes*** *to the provided input text. You can add, remove or replace individual words or punctuation characters, but* ***try to preserve the original meaning****. Return . . .*

B Examples for qualitative analysis

Example (1): *PR2.193* TREPAT does "light touch" edits more successfully.

Original text: faced with the backlash , the **Guardian** quickly – and very quietly – rowed back its initial certainty that its story was based on verified facts.

TREPAT: faced with the backlash , the guardian quickly – and very quietly – **it revised** its initial certainty that its story was based on confirmed data.

BERTattack: faced with the backlash, the **forward** quickly – and very quietly – rowed back its initial certainty that its story was based on verified facts.

Example (2a): *FC.184* TREPAT loses naturalness but violates meaning preservation less seriously.

Original text: the disaster artist (**film**). it had its world premiere at south by southwest on march 12, 2017. → the disaster artist (film) is **8** years old.

TREPAT: the disaster artist **movie debut** : it had its world premiere at south by southwest on march 12, 2017. → the disaster artist (film) is 8 years old.

BERTattack: the disaster artist (film). it had its world premiere at south by southwest on march 12, 2017. → the disaster artist (film) is **1** years old.

Example (2b): *FC.0* TREPAT repeats information unnaturally but retains meaning preservation.

Original text: → hannah and her sisters is an american **1986 film**

TREPAT: → hannah and her sisters is an american 1986 **american movie**

BERTattack: → hannah and her sisters is an american **1987** film

Example (3): *FC.128* TREPAT repeating entire phrases with the same information phrased slightly differently.

Original: colin kaepernick. kaepernick was selected by the **san francisco 49ers in the second round of** the 2011 nfl draft. → colin kaepernick starts for the **san** francisco 49ers.

TREPAT: colin kaepernick. kaepernick was selected by the **2011 nfl draft, kaepernick was selected by** the 2011 nfl draft. → colin kaepernick starts for the san francisco 49ers.

BERTattack: colin kaepernick. kaepernick was selected by the san francisco 49ers in the second round of the 2011 nfl draft. → colin kaepernick starts for the **special** francisco 49ers.

Example (4): *PR2.42* TREPAT reduces formality but sentence is still acceptable.

Original: we **failed to recall** that **they know nothing at all** about catholicism.

TREPAT: we **forgot** that they **don't have a clue** about catholicism.

BERTattack: we failed to recall that they **knows** nothing at all about catholicism.

Example (5): *PR2.115* TREPAT uses informality to create a bizarre tone, but maintaining the correct number for meaning preservation.

Original: sleipnir has **eight legs.**

TREPAT: sleipnir has eight **hooves!**

BERTattack: sleipnir has **six** legs.

C Annotation guidelines

Figure 3 presents the annotation guidelines provided to the linguists performing the manual evaluation of meaning preservation and language naturalness.

D Results for other victims

Tables 5, 6 and 7 show the full results of Experiment 3 for the BiLSTM, GEMMA2B and GEMMA7B victims, respectively.

Task overview

You have been asked to evaluate 199 short texts which have undergone two different modifications, labelled **Modified Text A** and **Modified Text B**, against the **Original Text.** Modifications are shaded green in the original text, and the replacement text is shown in red in the modified text columns (see Figure 1 below). Note also that some text may have been deleted completely in the process of modification, and they will only appear (in green) in the original text. For each text row in the provided spreadsheet, we would like you to tell us:

- Which of the two texts preserves the meaning and content of the original text better
- Which of the two texts sounds more *authentic*, take into consideration the original text's style and context - but also think about the text type (text types are explained below)

The text that does not look *authentic* is one that seems manipulated, fake or unnatural. You wouldn't trust the information that such text conveys.

Please indicate in the appropriate column by marking "A" for Modified Text A, "B" for Modified Text B, "Neither", or "Both". Mark your preference even if the difference is quite small (using "A" or "B") and avoid using other labels if you can. Please do not leave any fields blank. You can also add comments in the cell to the right of your decision.

Note that sometimes one variant achieves better authenticity and the other one preserves meaning better. See an example like this in Figure 2: while the replacement of "destroy" with "save" in text A sounds more natural and authentic than "knock" in text B, it also entirely changes the meaning of the original.

Text types

The texts you have been given to annotate cover 4 different categories:

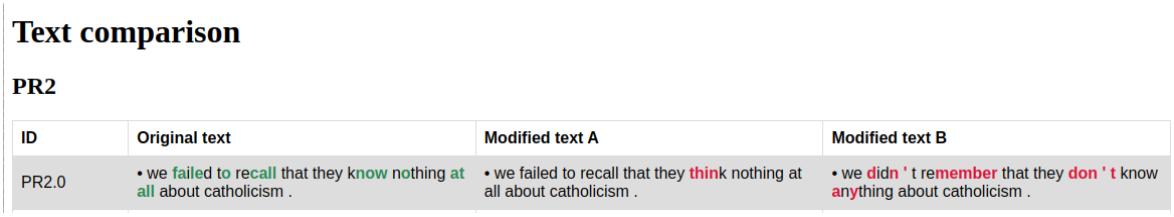
1) PR = Propaganda techniques (94 texts) in fragments of news articles.

2) FC = Fact-checking (57 texts) of isolated claims (includes the evidence from a knowledge base and a claim, separated by '~').

3) RD = Rumour detection (23 texts) in social media threads (messages separated by newline).

4) HN = Hyperpartisan news (25 texts) in online articles.

Note: do not try to assess the credibility of the text (e.g. if it is fake or real news) – this was already done by experts. However, knowing the task can help you decide which modifications are important. For the both metrics, the text type might have a different implication on *meaning preservation* or *authenticity* in the comparison you make with the original text. For example, authenticity of tone and style in "Propaganda" texts will be markedly different to "Rumour detection". Another example is the idea of meaning preservation - for "Fact-checking" texts, a small modification in a text which changes the content of the fact presented (e.g. a number) is considered a more serious violation of that metric compared to if the same modification happened in "Hyperpartisan news".

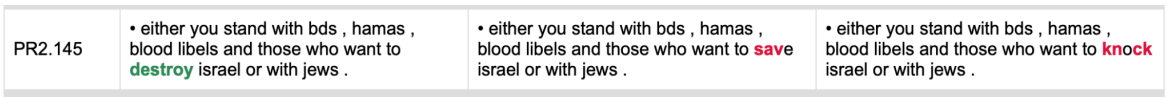


Text comparison

PR2

ID	Original text	Modified text A	Modified text B
PR2.0	• we failed to recall that they know nothing at all about catholicism .	• we failed to recall that they think nothing at all about catholicism .	• we didn ' t remember that they don ' t know anything about catholicism .

Figure 1



PR2.145	• either you stand with bds , hamas , blood libels and those who want to destroy israel or with jews .	• either you stand with bds , hamas , blood libels and those who want to save israel or with jews .	• either you stand with bds , hamas , blood libels and those who want to knock israel or with jews .

Figure 2

Figure 3: Annotation guidelines provided to the annotators.

Task	Prompt	BODEGA	Confusion	Semantic	Character	Queries
PR	BERT-ATTACK	0.3312	0.4760	0.7462	**0.9287**	37.0439
	F-BERT-ATTACK	0.3001	0.4543	0.7198	0.9123	32.7260
	BeamAttack	0.2382	0.3413	**0.7763**	0.8812	45.0463
	TREPAT-simple	0.2174	**0.7548**	0.5568	0.4704	18.5024
	TREPAT-full	**0.3493**	0.5457	0.7473	0.8399	21.8462
FC	BERT-ATTACK	0.2498	0.3333	0.7696	**0.9736**	45.8222
	F-BERT-ATTACK	0.1798	0.2420	0.7657	0.9695	41.9778
	BeamAttack	0.1054	0.1235	**0.8861**	0.9620	49.1580
	TREPAT-simple	**0.5176**	**0.7901**	0.7749	0.8297	17.5975
	TREPAT-full	0.4536	0.5753	0.8400	0.9352	27.3580
RD	BERT-ATTACK	0.0361	0.0627	0.5919	**0.9722**	48.8213
	F-BERT-ATTACK	0.0453	0.0892	0.5242	0.9718	47.0988
	BeamAttack	0.0261	0.0337	0.8189	0.9323	49.6643
	TREPAT-simple	**0.1840**	**0.2892**	0.7463	0.7967	41.2747
	TREPAT-full	0.1750	0.2145	**0.8597**	0.9396	42.7783
HN	BERT-ATTACK	0.0000	0.0000	0.0000	0.0000	50.0000
	F-BERT-ATTACK	0.0998	0.1525	0.6577	**0.9951**	44.0250
	BeamAttack	0.0000	0.0000	0.0000	0.0000	50.0000
	TREPAT-simple	**0.3856**	**0.4325**	0.9097	0.9795	33.5350
	TREPAT-full	0.1941	0.2075	**0.9399**	0.9948	43.5500

Table 5: Final evaluation results, showing the performance of TREPAT variants and baselines, applied to **BiLSTM** victim models trained for the four tasks. For each run, the mean BODEGA, confusion, semantic and character scores are included, as well as the number of queries.

Task	Prompt	BODEGA	Confusion	Semantic	Character	Queries
PR	BERT-ATTACK	0.2807	0.4087	0.7396	**0.9239**	38.6463
	F-BERT-ATTACK	0.2969	0.4399	0.7286	0.9218	31.7260
	BeamAttack	0.1627	0.2236	**0.7928**	0.8884	46.3317
	TREPAT-simple	0.1640	**0.7332**	0.4420	0.3712	22.2740
	TREPAT-full	**0.3062**	0.4928	0.7283	0.8304	24.3389
FC	BERT-ATTACK	0.2352	0.3185	0.7608	0.9701	46.3877
	F-BERT-ATTACK	0.1609	0.2247	0.7432	0.9623	43.8049
	BeamAttack	0.0927	0.1037	**0.9160**	**0.9743**	49.4864
	TREPAT-simple	**0.2998**	**0.5654**	0.6858	0.7451	29.4790
	TREPAT-full	0.2392	0.3333	0.7866	0.9075	37.8074
RD	BERT-ATTACK	0.0366	0.0699	0.5344	0.9784	48.7904
	F-BERT-ATTACK	0.0756	0.1614	0.4778	**0.9815**	44.6940
	BeamAttack	0.0274	0.0337	0.8419	0.9604	49.7133
	TREPAT-simple	0.1391	**0.2169**	0.7427	0.7896	42.3663
	TREPAT-full	**0.1484**	0.1783	**0.8626**	0.9531	44.0675
HN	BERT-ATTACK	0.0000	0.0000	0.0000	0.0000	50.0000
	F-BERT-ATTACK	0.1855	**0.2775**	0.6708	**0.9964**	37.4625
	BeamAttack	0.0000	0.0000	0.0000	0.0000	50.0000
	TREPAT-simple	**0.2016**	0.2250	0.9119	0.9825	41.3275
	TREPAT-full	0.1410	0.1500	**0.9436**	0.9958	45.0275

Table 6: Final evaluation results, showing the performance of TREPAT variants and baselines, applied to **GEMMA2B** victim models trained for the four tasks. For each run, the mean BODEGA, confusion, semantic and character scores are included, as well as the number of queries.

Task	Prompt	BODEGA	Confusion	Semantic	Character	Queries
PR	BERT-ATTACK	**0.2502**	0.3630	0.7386	**0.9305**	39.4878
	F-BERT-ATTACK	0.2453	0.3630	0.7298	0.9228	35.2668
	BeamAttack	0.1601	0.2163	**0.8080**	0.9058	45.9976
	TREPAT-simple	0.1555	**0.7476**	0.4140	0.3590	21.7428
	TREPAT-full	0.2379	0.3918	0.7217	0.8220	27.4255
FC	BERT-ATTACK	0.2442	0.3309	0.7596	0.9712	46.3086
	F-BERT-ATTACK	0.1471	0.2049	0.7433	0.9652	43.8864
	BeamAttack	0.1002	0.1111	**0.9227**	**0.9765**	49.5728
	TREPAT-simple	**0.3000**	**0.5481**	0.6948	0.7621	29.8642
	TREPAT-full	0.2487	0.3407	0.7962	0.9118	37.0370
RD	BERT-ATTACK	0.0363	0.0723	0.5190	0.9673	48.7639
	F-BERT-ATTACK	0.0510	0.1133	0.4612	**0.9770**	46.0313
	BeamAttack	0.0340	0.0410	**0.8692**	0.9528	49.5157
	TREPAT-simple	0.1275	**0.2193**	0.7102	0.7468	42.0578
	TREPAT-full	**0.1496**	0.1831	0.8589	0.9382	43.4988
HN	BERT-ATTACK	0.0000	0.0000	0.0000	0.0000	50.0000
	F-BERT-ATTACK	0.1348	0.2000	0.6763	**0.9968**	40.8225
	BeamAttack	0.0000	0.0000	0.0000	0.0000	50.0000
	TREPAT-simple	**0.2104**	**0.2325**	0.9196	0.9837	41.1775
	TREPAT-full	0.1199	0.1275	**0.9440**	0.9963	46.2150

Table 7: Final evaluation results, showing the performance of TREPAT variants and baselines, applied to **GEMMA7B** victim models trained for the four tasks. For each run, the mean BODEGA, confusion, semantic and character scores are included, as well as the number of queries.