Analyzing Robustness of Automatic Scientific Claim Verification Tools against Adversarial Rephrasing Attacks

JANET LAYNE, Boise State University, USA
QUDRAT E ALAHY RATUL, Boise State University, USA
EDOARDO SERRA, Boise State University, USA
SUSHIL JAJODIA, George Mason University, USA

The coronavirus pandemic has fostered an explosion of misinformation about the disease, including the risk and effectiveness of vaccination. AI tools for automatic Scientific Claim Verification (SCV) can be crucial to defeat misinformation campaigns spreading through social media channels. However, over the past years, many concerns have been raised about the robustness of AI to adversarial attacks, and the field of automatic scientific claim verification is not exempt. The risk is that such SCV tools may reinforce and legitimize the spread of fake scientific claims rather than refute them. This paper investigates the problem of generating adversarial attacks for SCV tools and shows that it is far more difficult than the generic NLP adversarial attack problem. The current NLP adversarial attack generators, when applied to SCV, often generate modified claims with entirely different meaning from the original. Even when the meaning is preserved, the modification of the generated claim is too simplistic (only a single word is changed), leaving many weaknesses of the SCV tools undiscovered. We propose T5-ParEvo, an iterative evolutionary attack generator, that is able to generate more complex and creative attacks while better preserving the semantics of the original claim. Using detailed quantitative and qualitative analysis, we demonstrate the efficacy of T5-ParEvo in comparison with existing attack generators.

Additional Key Words and Phrases: Neural Networks, Adversarial Attack, Scientific Claim Verification

1 INTRODUCTION

Scientific Claim Verification (SCV) tools are profoundly useful, particularly in a social climate where rampant misinformation pervades every media. As an example, the Coronavirus pandemic has fostered a plethora of false statements that have served to endanger the public health. These range from partial truths to outlandish suggestions: the vaccines are secretly used to implant microchips into the populace; the virus spreads through 5G. SCV tools have the potential to help protect the lay public from such onslaughts. However, understanding and predicting the veracity of scientific claims (such as those regarding medical subjects) is a particularly challenging natural language processing (NLP) task, due to the nuances of working with scientific jargon and the multitude of acronyms. To be useful in the real world, these tools need to discern scientific meaning from statements in many forms, including those that may not be ideally stated.

The most promising SCV tools in the literature are VERISCI [47], MultiVerS [49], and ARSJoint [54]. For a given claim, these tools return a decision label of either SUPPORT or REFUTE along with individual sources relevant to

Authors' addresses: Janet Layne, Boise State University, Department of Computer Science, Boise, Idaho, 83725, USA, janetlayne@boisestate.edu; Qudrat E Alahy Ratul, Boise State University, Department of Computer Science, Boise, Idaho, 83725, USA, qudratealahyratu@boisestate.edu; Edoardo Serra, Boise State University, Department of Computer Science, Boise, Idaho, 83725, USA, edoardoserra@boisestate.edu; Sushil Jajodia, George Mason University, Center for Secure Information Systems, Fairfax, Virginia, 22030, USA, jajodia@gmu.edu.

ACM 2157-6912/2024/5-ART
https://doi.org/10.1145/3663481

Table 1. Examples of ways to restate a scientific claim that are all semantically equivalent, and therefore should elicit the same response from a Scientific Claim Verification Tool.

Example Claim	Equivalent Claims
1. Antimicrobial agents are less effective due to the pressure of antimicrobial usage	2. Due to the pressure of antimicrobial usage, antimicrobial agents are less effective. 3. Pressure from antimicrobial use has reduced the effectiveness of antimicrobial agents. 4. Pressure from excessive use has caused antimicrobial agents to lose their effectiveness. 5. Excessive use of antimicrobials has weakened their effectiveness. 6. Effectiveness of antimicrobial agents is lower due to the pressure of antimicrobial usage.

the veracity of said claim. Each tool has a similar complex working schema: (i) retrieve relevant research paper abstracts in a database for a specific claim, (ii) identify particular statements in those abstracts that pertain to the veracity of the claim, and (iii) determine whether each statement supports or refutes said claim.

Studying the robustness of SCV tools is important to discover and resolve their weaknesses. Our experiments revealed a pronounced shortcoming in the existing SCV tools we tested: a claim restated in a different way, but with identical scientific meaning, may be classified differently than the original.

As an example, Table 1 shows several versions of a scientific claim. Each of these statements is semantically equivalent; however our experiments show that an SCV tool may return SUPPORT for some and REFUTE for others. Additional training to recognize a claim in all its possible forms can drastically improve SCV tools. Adversarial approaches offer an attractive strategy to assist this process via generation of attack claims that preserve semantic equivalence in terms of scientific meaning, but cause the SCV tool to reverse its classification. Ultimately, such generated adversarial claims can be used with the ground truth to further train the SCV tool and eliminate or mitigate the existing weaknesses.

The literature contains many attack models for generic NLP classification tasks. These models are divided into two categories: models that *a priori* generate attack sentences [13] (without considering the NLP classifier) and those that are classifier-driven. Classifier-driven models are better able to discover vulnerabilities specific to the considered classifier. Of these, two categories can be distinguished: (1) white-box [7], which uses information specific about classifier mechanisms (usually differentiable models are required), and (2) black-box [10, 16, 25, 41] that simply query the model without any extra information about the classifier. Due to the complexity of SCV tools that must first identify abstracts related to a claim, then further process them with multiple models, the use of a black-box attack model is the only option.

Our experiments show that the most advanced black-box attack models for generic NLP tasks do not perform well against SCV tools. They often generate modified claims with meaning entirely different from the original, or claims that are completely nonsensical with respect to either grammar or fact. We also show that, even for the successful attacks that preserve the original meaning, the modification is too straightforward, and fails to identify a diverse set of SCV tool weaknesses. A more complex set of attacks is needed to more broadly understand where improvements to SCV tools can be made.

To this end, we propose a new attack model, T5-ParEvo, which takes a scientific claim and rephrases it such that it is scientifically equivalent to the original, but will cause the SCV tool to classify it with the opposite label. T5-ParEvo is an iterative, evolutionary algorithm that uses a selection process driven by the SCV tool along with a semantic checker. This process iteratively improves the paraphrasing ability of the T5 language model to generate better and more diversified SCV tool attacks.

In this work, we evaluate the ability of T5-ParEvo to successfully attack VERISCI, MultiVerS, and ARSJoint. Moreover, we use T5-ParEvo to identify weaknesses of ChatGPT in accomplishing the SCV task with zero-shot learning.

In summary, contributions of the work include:

(1) The new T5-ParEvo model, which uses sophisticated training of the T5 paraphrasing model to generate multiple diverse attack claims to more comprehensively identify SCV tools' weaknesses. Importantly, these attack claims better preserve the scientific meaning of the original claim compared to the current state-of-the-art models.
(2) A framework for detailed analysis of success for adversarial attack models against SCV tools. This framework includes an extensive quantitative analysis of successful attack claims. This analysis enables in-depth evaluation of the diversity of attacks and whether scientific meaning is preserved in attack claims.
(3) An evaluation of T5-ParEvo, along with four state-of-the-art attack models according to the above framework.
(4) Ablation study showing the effectiveness of each component of T5-ParEvo.

This work is organized as follows: Section 2 provides a description of the related works to which we compare T5-ParEvo. Section 3 contains a detailed description of T5-ParEvo. Section 4 details the results of our experimental evaluation, and conclusions are provided in Section 5. In addition, we provide an Appendix that contains supplemental qualitative and quantitative analyses (A-C), reproducibility information (D), and a preliminary investigation of the robustness of ChatGPT as a potential SCV tool (E).

2 RELATED WORKS

In this section, we provide a comprehensive evaluation of related works in the field of automatic Scientific Claim Verification (SCV) tools and NLP Adversarial Attacks.

2.1 Scientific Claim Verification Tools

In the literature, there are several SCV tools [26, 36, 47–49, 54] that share a similar working structure, i.e., they take in input a claim, and return SUPPORT or REFUTE as an indication of whether the claim is true or false. In this paper, we focus our work on the three best-performing approaches with an available implementation that does not require paid cloud services: VERISCI [47], MultiVerS [49], and ARSJoint [54].

2.1.1 VERISCI. VERISCI [47] is one of the original SCV tools to evaluate scientific claims as supported or refuted by existing literature sources. Authors also developed the SCIFACT dataset (described in [47]), which is a set of 1409 scientific claims generated for multiple domains by scientific NLP experts, using sentences from scientific literature as sources. This dataset has been made available for use to train and test SCV tools, and authors created a LeaderBoard[1] for performance comparisons of many SCV tools on SCIFACT.

For a given scientific claim, VERISCI returns individual sentences (called rationale), along with a decision label for each rationale sentence of either SUPPORT or REFUTE. To accomplish this task, VERISCI uses three main submodules: AbstractRetrieval, RationaleSelection, and LabelPrediction. Given a claim, the AbstractRetrieval selects all the scientific abstracts that are related to the specific claim. This is done by hashing each abstract through TF-IDF encoding vectors and searching for each claim the top-k nearest abstracts by considering the distances between the claim and the abstract TF-IDF [1] encoding vectors. Once the top-k related abstracts are selected, the RationaleSelection module identifies rationale sentences for each abstract via a BERT model that determines whether a sentence of the abstract is related or not to the claim. Ultimately, the LabelPrediction

[1] https://leaderboard.allenai.org/scifact/submissions/public

module, given a claim, an abstract, and all sentences for the specific abstract, determines using a RoBERTa transformer network whether the claim is supported, refuted by the abstract, or whether there is no information. The last two modules are trained separately with the SCIFACT dataset.

2.1.2 MultiVerS. This is a newer scientific claim verification tool created by the authors of VERISCI. MultiVerS is the current leader according to AllenAI Leaderboard on the SCIFACT dataset. MultiVerS (previously named Longchecker [50]) returns a similar output as that for VERISICI; however, where VERISCI predicts the label of a rationale as SUPPORT or REFUTE using only the rationale itself, MultiVerS was expanded to include the full abstract in which the rationale statements are found as context to determine the label for the statement. This change is based on a finding during the development of VERISCI that the model showed a tendency to perform poorly on rationale statements that were sensible only with outside context. In order to learn from the entire abstract, a long-document transformer, Longformer [3] trained on the S2ORC corpus was utilized. Unlike VERISCI, MultiVerS uses a multitask loss for rationale selection and label prediction, rather than training these as separate modules. For candidate abstract retrieval, MultiVerS uses the Vert5Erini retrieval system. MultiVerS was fine-tuned on a large set of supervised, out-of-domain claims from the FEVER [44] fact-checking dataset, along with weakly-supervised in-domain data from two datasets: EVIDENCEINFERENCE [6, 23] and PUBMEDQA [17]. The SCIFACT claim/evidence pairs were then used to further fine-tune, along with negative examples (claim/abstract pairs with no relationship). MultiVerS outperformed ParagraphJoint [26] and Vert5Erini[36], thus we will not include these approaches for comparison in our experiments.

2.1.3 ARSJoint. This model [54], jointly trains on all three tasks (abstract retrieval, rationale selection, stance prediction) in the framework, with a regularization based upon the divergence between the sentence attention of abstract retrieval and the output of rationale selection. This regularization allows information exchange among these two tasks. Like MultiVerS, ARSJoint uses the word representation of the entire abstract, but, unlike MultiVerS, stays within the 512 token limit of the BioBERT [22] transformer.

2.2 NLP Adversarial Attack Tools

Although there are several adversarial attack models, these are largely meant for generic NLP classification tasks, rather than for scientific claim verification tools. There exist two main types of attack models for generic NLP classification tasks: *a priori* models that generate attack sentences with no consideration of the classifier, and those that are classifier-driven. Classifier-driven models are superior for our task, and consist of white-box, black-box, and hybrid models.

White-box models [28, 42] utilize gradients, structure, or parameters of the target model to guide modification of original samples to adversarial ones. Scientific claim verification tools are more complex than simple text classification models, as they are often not differentiable (at least those with best performance). This is because the classification decision is made according to retrieved abstract documents. This implies that white-box approaches such as Relgan [33] using the Gumbel softmax trick [14] cannot be used with scientific claim verification tools.

Black-box attacks are classifier driven, but unlike white-box attacks, have no access to specific information of the classifier, including gradients, parameters, and structure. As such, these attacks can only query the classifier as an oracle to guide generation of adversarial claims.

Hybrid models [20, 30] utilize a transfer method, wherein a surrogate white-box model is used with gradients to direct generation of adversarial examples. These are then used as black-box attacks on different target models. More specifically, a surrogate differentiable model that approximates the black-box NLP model is created and used to direct attacks. However, scientific claim verification tools contain an information retrieval component to search for relevant abstracts which renders them non-differentiable. Approximating SCV tools that depend on a retrieval process from extensive abstract database with a surrogate differentiable model is not realistic, and is in

opposition with the research direction of SCV tools. Thus, we consider such a transfer method with surrogate models unsuitable for our problem.

Several black-box (and white-box) models simply exchange words with synonyms using dictionaries [24, 40]. More sophisticated techniques generate synonyms by perturbations of word embeddings (see Glove [9]) or use language models to score the generated perturbations [2, 29]. To create more complex attacks that do not simply change words in a sentence, [27] uses phrase-level insertion and deletion; however, it produces unnatural sentences. Other works [11, 15] focus on a manual strategy to attack the model, but require effort from the analyst. BAE [10] creates attacks by adding, changing, and removing words. For this reason, we have chosen this model as one, among others, for comparison to our method.

Unfortunately, all state-of-the-art automatic adversarial attacks for text procedures, when considered in the context of automatic scientific claim verification, have similar limitations: (i) they often produce attacked claims with a meaning completely different from the original claim, and (ii) among the few attacked claims that do preserve the meaning, the attack consists only of swapping a single word. There is a distinct lack of sophistication in strategy, even for approaches like BAE.

In our experiments, we observe in the state-of-the-art attack models a lack of understanding of scientific terms, a failure even to generate coherent synonym exchanges given the sentence context, and modifications of the original claims that are too simplistic to comprehensively attack SCV tools. In the following, we describe the approaches and limitations of the most effective black-box attack techniques to be compared with our T5-ParEvo procedure.

2.2.1 TextFooler. TextFooler is a model specifically designed to generate adversarial text in order to fool text classification, and textual entailment tasks [16]. The attacker queries the target model to get feedback for training in the form of predictions corresponding to confidence scores. To generate attacks, the model begins by ranking word importance in the original claim. A BERT-based model is used to determine the words that are most influential in the outcome of the attack. This is then used to select the words with the greatest influence to alter in order to minimize the number of alterations, because it is expected that minimizing alterations to the claim will best preserve semantic similarity. Once these words have been chosen, the word transformer is used to replace them. This consists of 3 steps:

(1) Synonym Extraction: a set of candidates for all possible replacements is generated with the N closest synonyms (based on cosine similarity of word embeddings from Mrksic et al. 2016).
(2) POS Checking: the set of candidates are filtered for only those with the same part-of-speech (POS).
(3) Semantic Similarity checking: The Universal Sentence Encoder (USE) is used to generate sentence embeddings, which are then checked for cosine similarity.

If this set of candidates contains claims which result in a successful attack, the candidate word with the highest semantic similarity is chosen. If, however, no new claim is yet successful, this process is repeated to transform the next most important word.

This approach, along with all the others described here, suffers from a major weakness in that only a single attack is generated for each original claim, asserted to be the best candidate attack. Additionally, it operates via word replacement, and by prioritizing the replacement of important words, technical words can be replaced. This often leads to a loss of meaning equivalence in scientific claims. There are a large number of terms that have different meanings when used in a technical context than in general speech. Attempts to exchange biological terms, for instance, with the synonym that makes sense in a general speech context, can result in incoherent claims. Further, though attempts are made to preserve part-of-speech and semantic similarity, altering single words can result in a loss of context and generate nonsensical sentences with respect to grammar.

2.2.2 CheckList. Checklist is a model for which the intended use is as a generator of test cases for NLP tasks to identify failures [41]. Checklist generates sentences which test specific capabilities of NLP models, including: Negation, Vocabulary, and Named-Entity Recognition (NER). Test sentences are generated to conduct testing of each of the following types: minimum function test, invariance test, and directional expectation test. Tests can be generated from existing statements (as in our use-case) using RoBERTa suggestions combined with WordNet categories to guarantee context-based synonyms. Again, claim generation at the word-replacement level creates issues with scientific claims. Our experiments also showed substitution of words with antonyms rather than synonyms, or simply words that were not remotely related, let alone synonymous with the word being replaced. This resulted in either a direct reversal of the meaning of the claim, or at least a change in the meaning, nullifying its validity as a successful attack. Moreover, this model showed very little success in attacking the scientific claim verification tool from the outset.

2.2.3 Bert Attack. Bert Attack [25] is an attack model that, by substituting words of an original sentence, generates new sentences. Differently from TextFooler and CheckList, the substitution of words employs BERT to ensure that for each substitution the semantic role of the word in the entire context is considered. Though this theoretically should improve performance with respect to the previously described weaknesses, we observed that the model still failed to understand the complexity of technical terminology, and also allowed substitution of words that were not synonyms but antonyms that fit well in the context of the sentence. This directly reversed the meaning of the claim, and as such was not a legitimate attack against the SCV tool.

2.2.4 BAE. BERT-based Adversarial Examples is an attack model that generates new sentences from existing ones differently than the previously described related works [10]. Briefly, the model takes an existing sentence and masks one of two parts of that sentence: either a given word or directly to the left or right of the given word, which results in an insertion to the sentence rather than a substitution. A BERT-based language model (LM) is then used to replace the mask based on the contextual meaning of the original sentence. Tokens to be masked are chosen based upon a higher Universal Sentence Encoder-based similarity score. For insertions, a POS check is also used to filter candidates. The token chosen is then that which best attacks the model: either causes misclassification or lowest prediction score. Like TextFooler, these perturbations are applied iteratively in order of decreasing token importance. In contrast with the previous three described approaches, BAE has the ability to add words to a sentence, rather than simply substitute existing words. This may represent an improvement over the other related works, however, there is still a lack of understanding of scientific jargon. Also, the adding operation is performed iteratively by adding a word followed by a check at each step for the consistency of the sentence. This results in claim sentences with the same structure as the original. As with TextFooler, this model struggles with words that have relevant scientific meaning that is different than the standard meaning in most sentences (e.g. "**cell**").

3 T5-PAREVO

In this section, we describe our attack claim generator, T5-ParEvo, which automatically generates adversarial claim attacks for a generic black-box scientific claim verification tool. Given our black-box assumption, during the attack generation process, it is only possible to use the tool as an oracle, i.e., one can only query the tool to classify the claim, but no other information is provided.

Then, given a scientific claim C_{orig} and a black-box scientific claim verification tool CV, our attack claim generator outputs a paraphrased modification C_{par} of the original claim C_{orig} such that (i) C_{par} is semantically equivalent to C_{orig} and (ii) CV classifies C_{par} differently than C_{orig}. As an example, consider the original claim: C_{orig} -"Antimicrobial agents are less effective due to the pressure of antimicrobial usage." which is classified as true by the scientific claim verification tool. Now consider the two paraphrased modifications: C^1_{par} -"Antimicrobial agents

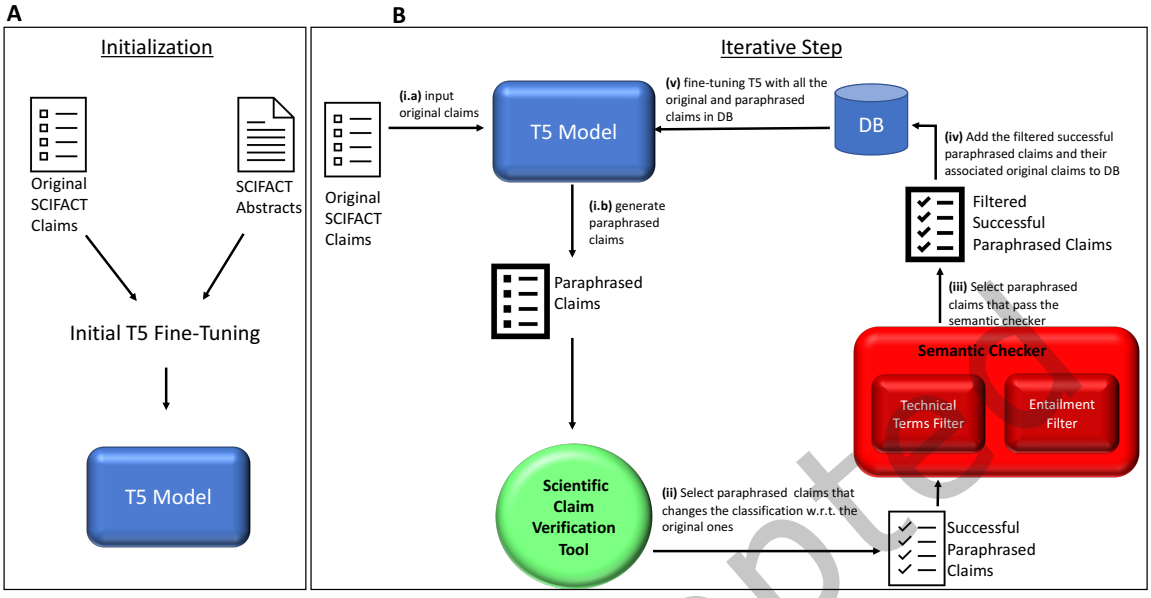

Fig. 1. Schematic of **T5 – ParEvo**. (A) The initial fine tuning of the T5 model. (B) The subsequent iterative fine tuning of the T5 model.

are more effective due to the pressure of antimicrobial usage" and C^2_{par} - "Due to the pressure of antimicrobial use, antimicrobial agents are less effective." Both generated claims reverse the decision of the claim verification tool to false. However, we consider only C^2_{par} as a valid adversarial claim attack; C^1_{par} alters the semantic meaning of C_{orig} and therefore we do not consider it a valid attack.

In summary, given a black-box scientific claim verification tool CV and a set L of claims, our attack model generator T5-ParEvo generates a set of adversarial claim attacks for each claim C in L simultaneously.

Now we provide a high-level view of our adversarial attack claim generator T5-ParEvo. T5-ParEvo is an evolutionary procedure that, at each iteration, improves the ability of an autoregressive generative language model, such as the T5 transformer, to generate (given a set of claims) adversarial claim attacks. Intuitively, it is possible to consider an autoregressive generative language model (e.g., T5) as a function that takes as input a claim and returns several modifications of it. The challenge that T5-ParEvo accomplishes is to train such a function in a way that the modifications of the claim are semantically equivalent to the original while simultaneously reversing the classification outcome of the scientific claim verification tool.

In Figure 1, we provide an overview of the attack model generator T5-ParEvo, which consists of an initialization step (Figure 1.A) and the iterative evolutive procedure (Figure 1.B). The initialization step consists of an unsupervised fine-tuning of the T5 model with both the claims for the attack generation and the related abstracts. Though T5 is already trained on a large corpus, this step is crucial to impart improved usage of scientific language. This initialization is critical for further iterative steps.

The iterative step of the T5ParEvo, as depicted in Figure 1.B, consists of (i) generating several paraphrased claims from the original using the current T5 model, (ii) selecting all the paraphrased claims that change the

Algorithm 1 T5-ParEvo

function T5 − ParEvo(Scientific Claim Verification Tool CV, Set of claims L, Number of paraphrased claim to return h, Iterations k)
  Initialize $DB = \emptyset$
  Initialize $T5$
  for all $i = 1 \ldots k$ **do**
    $IS = \{(C_{orig}, C_{par}) | C_{par} \in T5(C_{orig}, h), C_{orig} \in L\}$
    $SS = \{(C_{orig}, C_{par}) | CV(C_{orig}) \neq CV(C_{par}), (C_{orig}, C_{par}) \in IS\}$
    $SCS = \{(C_{orig}, C_{par}) | \textbf{SemanticChecker}(C_{orig}, C_{par}) = True, (C_{orig}, C_{par}) \in SS\}$
    $DB = DB \cup SCS$
    Initialize $T5$ and fine-tune it with the pairs in DB
  end for
  return DB
end function

function semanticChecker(Original claim C_{orig}, Paraphrased claim C_{par})
  Compute the sets $TT(C_{orig})$ and $TT(C_{par})$ of technical terms (one or more words) and their number of occurrences in C_{orig} and C_{orig}
  Let $\models$ the transformer based model for sentence entailment
  if $TT(C_{orig}) = TT(C_{par}) \wedge C_{orig} \models C_{par} \wedge C_{par} \models C_{orig}$ **then**
    return True
  end if
  return False
end function

classification outcome of the scientific claim verification tool with respect to the original claims, (iii) further selection of the claims that pass our defined semantic checker, (iv) add the finalized filtered paraphrased claims and their associated original claims to the database DB, and (v) fine-tuning the T5 model with all the pairs of original and paraphrased claims contained in DB.

An example of the effects of the iterative procedure is shown in Figure 2. As shown in the figure, some paraphrased claims will fail to change the decision of the scientific claim verification tool, and therefore will not proceed in the process. Then, successful claims will further be subject to the Semantic Checker, which will remove those for which scientific terms have been changed, or for which the original and paraphrased claims do not entail one another. Claims that pass through this filter will be used to further fine-tune the model. This is performed for each of the original claims, for 50 iterations, or until no new paraphrased claims are generated. Please note that the first filter operation (action ii in Figure 1.B) enforces the fact that the paraphrased claims change the classification result of the scientific claim verification tool w.r.t. to the original claim, while the second filter operation (action iii in Figure 1.B) enforces that the change in classification of the scientific claim verification tool is due to effective weaknesses of the scientific claim verification tool (i.e., the claim is restated in such a way as to be misunderstood by the tool) rather than a semantic change of the paraphrased claim in comparison with the original - which would render the tool correct in its change of classification (e.g., C^1_{par} in the above example).

This model has two main advantages w.r.t. to the state of the art. First, it learns the weakness of the scientific claim verification tool from all the available claims rather than only one claim at a time. The specific weakness identified for a claim is potentially generalized and extended to other claims. Second, the semantic checker filter is designed specifically for scientific claims. It recognizes technical terms and focuses on the entailment task between two sentences. Third, our model largely uses transfer learning to improve the evolution of the attacks and judge their semantics.

In Algorithm 1 we provide the entire T5-ParEvo procedure and in the following sections, we provide a detailed explanation of T5-ParEvo and its semantic checker function, along with a description of the initial fine-tuning of the T5 model and the entailment module.

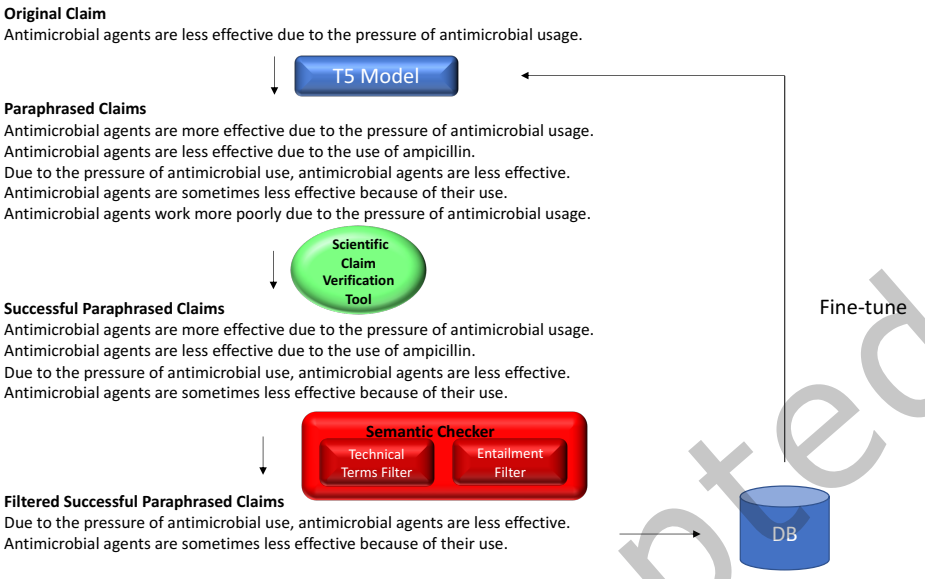

Fig. 2. Example of a single iteration of the T5-ParEvo fine tuning. Paraphrased claims are manufactured examples to show the effect of the feedback and filtering from the Scientific Claim Verification Tool and Semantic Checker, respectively.

3.1 Generating Paraphrased Claims and T5-ParEvo Initialization

T5-ParEvo uses the T5 transformer model to generate claims via the paraphrasing task. Transformers are neural networks models characterized by their use of attention layers [46]. They are often used as language models - trained in an unsupervised way over a large text corpus - that learn the (conditional) distribution of the text in a corpus. They are often used for language translation, but are useful for a multitude of NLP tasks. When used as language models, they use the attention layers to identify connections (context) between words in a sentence. This helps to identify the conditional distribution of the sentences and produce a meaningful representation of the text. In particular, such representations can be used for supervised classification (e.g., to check if a sentence entails another one), comparison of representations to establish semantic similarities between sentences, etc. Translation model transformers can be used to translate text from one language to another, but also to summarize or paraphrase text in the same language. Popular models using transformers are BERT [46], ALBERT [21], RoBERTa [31], GPT [37], and T5 [38].

The paraphrasing task, given an input text, consists of restating that text such that the transformed and the original have the same semantics. To accomplish this task, a T5 model can be fine-tuned with a specific paraphrasing dataset after initial unsupervised training on a large corpus. A paraphrasing dataset contains pairs of texts: an original text and its paraphrased one. Fine-tuning is accomplished via maximizing the likelihood of generating the paraphrased text, given in input the original text.

In [13], authors propose syntactically controlled paraphrase networks, which improved the performance of T5 at a paraphrasing task with respect to the number of diversified and semantically equivalent generated sentences.

In principle, paraphrasing models can be used directly to generate adversarial attacks. However, in the absence of guidance from the scientific claim verification tool, simple paraphrasing is unlikely to be successful. As such,

in this work we use a procedure to evolve a paraphrased model using guidance obtained from the result of the black-box model under attack. This is confirmed in Figure 3, where iterations of our method resulted in not only increased success in the attack, but also increased diversity of attacks for each claim.

Due to the existence of multiple transformer-based paraphrasing models, in Appendix A3 we evaluated several models, including: T5base, T5PAWS (our base paraphraser, described in Section 3.1), bart_{eugene}, ProtAugment, $\text{bart}_{stanford}$, mt0, alpaca, and Primera. Our evaluation uses the paraphrased claim produced by each model and the quantitative analysis described in Section 4.2 to evaluate the abilities of such models to preserve the semantics and generate grammatically correct and syntactically different paraphrased sentences. Such analysis shows that the T5 model is the most appropriate for our requirements.

Thus, we use a T5 model already trained with google PAWS [53], available through the HuggingFace repository[2].

We fine-tune the T5 model with the denoising training procedure [38] (unsupervised) using the SCIFACT dataset claims and abstracts corpus (Figure 1.A and Line 3 of Algorithm 1). Specifically, each abstract is split into atomic sentences for a total of 45 952 sentences. We masked 15% of tokens, and trained the model for 3 steps, for a total of 3 different masking schemes. To establish a robust masking procedure that avoids partial tokenization of scientific terms which would result in separate masking for multi-word terms (e.g. stage IV carcinoma), we use a scispaCy en_core_sci_scibert [3] Named Entity Recognition model for sentence tokenization.

3.2 Semantic Checker

The function **semanticChecker** in Algorithm 1 (Lines 13-20) is our proposed heuristic to determine if two claims C_{orig} and C_{par} are semantically equivalent. First, the function identifies the technical terms (Line 26) TT_{orig} and TT_{par} in the two claims. These technical terms can be composed of one word or multiple. Technical terms are identified using named entity recognition (NER), a sub-task of information extraction that identifies plain-text entities composed of one or more words [32]. One popular library used for named-entity recognition is spaCy[4], which can be trained to recognize entities of a specific domain. In the context of scientific or biomedical statements, in [39] authors created hybrid approaches with spaCy and specific domain training datasets to recognize scientific entities. In the scispaCy[5] repository four pretrained spacy NER models can be found: (1) $en\_ner\_bionlp13cg\_md$, (2) $en\_ner\_jnlpba\_md$, (3) $en\_ner\_craft\_md$ and (4) $en\_ner\_bc5cdr\_md$. These models are each trained on a different dataset and are specific for biology, scientific and clinical text. Each demonstrated greater than 70% F1 score for the NER task. The result of each of these four models may be different when applied to a specific claim. Our semantic checker utilizes all four to maximize recognition of technical terms.

One of the first conditions to verify whether two claims are semantically equivalent is that TT_{orig} and TT_{par} are the same, i.e., in the paraphrasing procedure, none of the technical terms was modified or lost. We allow only two suffix changes to technical terms: -ed and -s.

As a second condition the function uses a RoBERTa model $\models$ pretrained for the natural language inference task, to ensure two-way entailment, i.e., $C_{orig} \models C_{par} \wedge C_{par} \models C_{orig}$. This is a typical strategy in logic to establish logical equivalence. The natural language inference task consists of determining whether a hypothesis written in plain text is true given a premise written in plain text. If so, the premise entails the hypothesis. For this task, transformer models such RoBERTa [31] are trained with corpus datasets specific for the inference task. The most frequently used datasets for Inference are Stanford Natural Language Inference (SNLI) Corpus [4], Multi-Genre Natural Language Inference (MultiNLI) Corpus [51], and SciTail Corpus [18]. For the entailment module, we adopt a RoBERTa model[6] already pretrained on the MultiNLI corpus which showed 90% accuracy at the entailment

[2]https://huggingface.co/seduerr/t5-pawraphrase
[3]https://allenai.github.io/scispacy/
[4]https://spacy.io/
[5]https://allenai.github.io/scispacy/
[6]https://huggingface.co/roberta-large-mnli

task. This was selected because it is one of the largest corpora, and contains multiple genres of texts. We further fine-tune this model for our entailment task specific to scientific claims with the SciTail Dataset[7]. In principle, the entailment check should be sufficient; however, as technical terms are often not well understood even by the trained RoBERTa model, the upstream condition preserving technical words is also required. Both the initial fine-tuning tasks are essential for providing specific domain knowledge to the T5-ParEvo procedure. This is empirically shown in the ablation study described in Section 4.5.

3.3 Iterative Evolutionary Procedure

T5-ParEvo generates a set of attacks in the form of (C_{orig}, C_{par}) via an iterative evolutionary procedure. At each iteration, the T5 model generates for each original claim h modified (paraphrased) claims (Algorithm 1 Line 5). These paraphrased claims are then used to challenge the black-box scientific claim verification tool. Each claim that successfully reverses the classification outcome (Line 5) is then passed to the semantic checker described above. Those claims which preserve entailment and technical terms, i.e., the SemanticChecker function returns True (Line 7), are then added to a database DB of training claims (Line 8), subsequently used to fine-tune the T5 model (Line 9). Through such iterations, the ability of the T5-model to create from an original claim C_{orig}, a paraphrased claim C_{par} evolves in order to better generate successful attacks. This also allows the model to generalize successful strategies from one claim to the others. This is markedly different from the related works, which operate on a single claim with feedback from the target model about that claim only.

More specifically, T5-ParEvo takes in input a scientific claim verification tool CV, a set of claims L, and two numbers h and k. It starts by initializing the set DB of attacks to the empty set (Line 2) and the T5 model (described in Section 3.1). At each iteration the T5-ParEvo uses the T5 model fine-tuned at the previous iteration to generate for each claim C_{orig} in SC, h different paraphrased claims (denoted as C_{par}). All the pairs of original claims and paraphrased claims are stored in the initial set IS (Line 5). The method selects all the pairs (C_{par}, C_{orig}) in IS such that classes assigned to them by CV are different (i.e., $CV(C_{orig}) \neq CV(C_{par})$) and stores them in successful set SS (Line 6). All pairs in SS are then filtered by the function **SemanticChecker** to obtain the set Semantic Checked Set SCS (Line 7). Then, the pairs in SCS are added to DB (Line 8), and DB is used to fine-tune a new initialized T5Par model (Line 9) that is used in the next iteration.

Note that T5 model at each iteration can be fine-tuned with multiple pairs of claims for each original claim in L (see Line 5 of Algorithm 1). The number of pairs for each original claim at each iteration depends upon the filtering process done in Lines 6-7. In addition, at each iteration, the pairs of claims used for the fine-tuning of the T5 model are the union of the new pairs generated for the current iteration and all those generated at the previous iterations, this increases the chances, at each iteration, the T5 model is fine-tuned with multiple pairs of claims for each original one in L. The T5-ParEvo function computes a total of k iteration (Line 4) and k is set to 50, with manual stopping if several iterations pass with no additional unique attacks produced. This was observed for MultiVerS and ARSJoint (Appendix A2) Once all iterations are completed T5-ParEvo returns the set DB.

T5-ParEvo is an evolutionary algorithm [35] because: (i) the initial population of individuals is generated by paraphrasing the original claims (Line 5), (ii) the selection process is performed by the semantic checker and the scientific claim verification tool CV (line 6 and 7), and (iii) the crossover/mutation operations are both effectuated by fine-tuning first the T5 model with the successful pair that survived the selection process (Line 9) and then again generating the paraphrased claims (Line 5).

4 EXPERIMENTS

In this section, we provide an empirical comparison of T5-ParEvo with several state-of-the-art text attack generators. Our comparison is performed by analyzing the results of each text attack generator in two ways:

[7] https://allenai.org/data/scitail

quantitatively and qualitatively. In the quantitative analysis, we define several machine-computable heuristic measures to evaluate the paraphrased claims with respect to grammar, syntactic (not necessarily semantic) differences, and preservation of meaning. This quantitative analysis is performed for T5-ParEvo and all four related works on VERISCI, and for T5-ParEvo along with TextFooler (the best performing related work) for MultiVerS and ARSJoint.

For our qualitative analysis, the authors (including a scientific domain expert[8]) study the attacks generated by each text attack generator on VERISCI, and create a logic schema to categorize the quality of the comments in three categories (high/medium/low). Additionally, we characterize attack strategies for each model, along with types of mistakes that render an attack invalid. This set of quantitative and qualitative analyses on their own are a contribution of our paper, such that they present a valuable evaluation metric to be used for assessment of future text attacks on scientific claim verification tools.

In the following sections, we provide a detailed description of our experimental settings, quantitative analysis, and qualitative analysis. Moreover, we conduct experiments on the creation of a classification model that distinguishes between original claims and paraphrased attack claims, and discuss the benefit of the initialization and the iterative step of T5ParEvo.

4.1 Experiment Setting

For our experiments we used three scientific claim verification tools: VERISCI, MultiVerS, and ARSJoint. Those listed are the three top-performers[9] that have an available implementation that does not require any cloud service to run. The F1-scores for verification of a scientific claim on the SCIFACT dataset test are as follows: 0.47 for VERISCI, 0.7248 for MultiVerS, and 0.7123 for ARSJoint. In addition to those models, in the Appendix (A5) we describe preliminary tests on ChatGPT [34].

For each of the scientific claim verification tools, we use a subset of the claims of the SCIFACT dataset to attack the claim verification tool. Namely, we consider all the claims where the verification tool returns true or false (all claims where the verification tool is unable to reach a decision are discarded). Then, we remove claims that were initially misclassified. This step is not required, however, we removed them to ensure that each attack represents a weakness of the verification tool. A previously misclassified claim that is then correctly classified upon rephrasing is difficult to assess as a weakness of an SCV tool.

As related works for T5ParEvo we consider TextFooler, BAE, Bert Attack, and Checklist. An important difference exists between the related works and our model. For each of the related works, attacks are continuously generated until one is successful, then that single attack is returned. If no successful attack can be generated, the original claim is returned. However, T5-ParEvo at its core is a paraphrasing model, and as such generates many paraphrased versions of each original claim, which are then used to attack the scientific claim verification tools. Some of these claims will be successful in attacking the SCV tool, while others are not. In contrast with the related works, for a single original claim, T5-ParEvo can return several paraphrased sentences that are successful attacks. When describing results, we, therefore, refer to "unique attacks" to make useful comparisons between our model and the related works. As an example, for a single original claim, T5-ParEvo may generate 10 paraphrased claims, 7 of which are successful in attacking the SCV tool. This would be counted as 1 "unique attack".

For T5-ParEvo implementations, we set the number of iterations k to 50, with manual stopping in the event that the number of unique successful attacks stops increasing. For the other models, we used the default parameters suggested in their documentation.

Table 2. Quantitive analyses between T5-ParEvo and other attack methods applied to VERISCI.

		T5-ParEvo	TextFooler	BAE (BERT Adverserial)	Bert Attack	Check List
Number of Unique Attacks		155	229	242	219	16
Number of Unique Attacks Passing Semantic Checker		155	88	29	34	4
Scientific Term Preserved (%)		100.00	71.00	61.00	62.00	76.00
Two-way Entailment Preserved (%)		100.00	78.83	38.00	61.34	90.77
N-gram Manhattan Distance (mean)	Uni-gram	5.10	2.00	2.20	2.23	0.50
	Bi-gram	6.57	3.21	3.19	3.24	0.56
	Tri-gram	7.59	3.39	3.86	3.90	0.73
Avg. number of Grammatical Errors		0.35	0.60	0.76	0.89	0.50

4.2 Quantitative Analysis

Our quantitative analysis includes the following metrics to evaluate each paraphrased claim: (i) preservation of scientific terms, (ii) entailment, (iii) syntactic (not necessarily semantic) differences w.r.t. the original claims, (iv) grammatical correctness, and (v) readability.

In the following, we give an overview of the defined metrics, then provide a subsection explaining each in detail. To measure whether the semantics of the original claims are preserved in the paraphrased claims, we propose three measures: number of attacks passing the semantic checker, scientific term preserved (%), and two-way entailment (%). Then we measure how syntactically different the paraphrased claims are from the original using a N-gram Manhattan distance (mean). Last, we use the average number of grammatical errors to measure the grammatical correctness of the paraphrased claim along with several readability measures to evaluate the readability.

Table 2 shows the results of the above quantitative measures (excepting readability) for T5ParEvo and the related works using VERISCI. Table 3 shows these same results for T5ParEvo and TextFooler (the best performer of the related works in Table 2) for MultiVerS and ARSJoint. Please note that, in contrast to the related work, T5-ParEvo returns multiple paraphrased attack claims per original claim. As such, we report in the tables the number of successful unique attacks, i.e., the number of original claims for which there exists *at least one paraphrased attack claim* for that original claim. Last, we describe in Table 4 four readability measures for the paraphrased attacks generated for VERISCI.

[8]molecular biologist
[9]https://leaderboard.allenai.org/scifact/submissions/public

Table 3. Quantitive analyses between T5-ParEvo and TextFooler applied to MultiVerS and ARSJoint.

		T5-ParEvo (MultiVerS)	TextFooler (MultiVerS)	T5-ParEvo (ARSJoint)	TextFooler (ARSJoint)
Number of Unique Attacks		114	147	139	212
Number of Unique Attacks Passing Semantic Checker		114	61	139	72
Scientific Term Preserved (%)		100.00	73.67	100.00	63.39
Two-way Entailment Preserved (%)		100.00	69.93	100.00	57.26
N-gram Manhattan Distance (mean)	Uni-gram	3.99	2.13	4.40	2.09
	Bi-gram	7.624	3.47	6.94	2.10
	Tri-gram	8.99	3.36	8.52	3.43
Avg. number of Grammatical Errors		0.39	0.59	0.37	0.66

Given these measurements, we demonstrate that T5ParEvo w.r.t. the related works creates superior attacks that better preserve the semantics of the original claims, which are more grammatically correct and syntactically more different from the original claim. Despite the large syntactic differences between the original claims and T5-ParEvo attacks, the level of readability remains comparable with the other attack models. We detail the results for each measure for T5-ParEvo and the related works below.

4.2.1 Measure 1: Number of Unique Attacks (Passing the Semantic Checker). As previously described, because T5-ParEvo returns multiple candidates per original claim, where the related works return only one attack per original claim, we report in Table 2 and Table 3 the number of unique attacks as the number of original claims for which there exists *at least one* successful paraphrased attack claim. For the related works, the number of unique attacks is the same as the number of attacks.

For the task of attacking scientific claim verification tools, it is vital that the paraphrased attack claims preserve the semantics of the original claims. Thus, we report not only the number of unique attacks but those passing the semantic checker, i.e. the number of original claims for which *at least one paraphrased attack claim* passes the semantic checker defined in Section 3.2. Please note that because T5ParEvo integrates the semantic checker, the number of unique attacks is the same as the number of unique attacks passing the semantic checker. For the related works, these two numbers are different, as not all their attacks pass our semantic checker. As described in section 3.2, if the paraphrased attack passes the semantic checker, it preserves the technical terms from the

original claim and both the paraphrased attack and the original entail one another (two-way entailment). We also decompose the results from the semantic checker into each individual measure, described in Section 4.2.2 and Section 4.2.3.

From Table 2 and Table 3 we observe that, though many of the related works have a larger number of unique attacks than T5ParEvo, most did not pass the semantic checker. Considering only these passing claims, T5ParEvo outperforms the second best by nearly double. This result is strongly confirmed in our qualitative analysis, in which a domain expert deemed that over 85% of successful attacks from the related works were not valid.

4.2.2 Measure 2: Scientific Term Preserved (%). Given a pair of claims (C_{orig}, C_{par}) such that C_{orig} is the original claim and C_{par} is the generated claim, the percentage of scientific terms preserved is defined as follows:

Let $TT(C_{orig})$ be the set of scientific terms in C_{orig} as identified by the NER model described in Section 3.2 and $TT(C_{par})$ be the set of scientific terms in C_{par}. Linear sum assignment[10] is used to construct a bipartite graph between $TT(C_{orig})$ and $TT(C_{par})$, with the goal of finding a complete assignment of terms between the sets. This is treated as a binary problem where each edge is given a match value of 1 if they are a match, and a 0 otherwise. An allowance for matching is made for suffixes -ed or -s. The preservation value $PRST(C_{orig}, C_{par})$ for a pair (C_{orig}, C_{par}) is calculated as $\frac{linear\_sum\_assignment(C_{orig}, C_{par})}{|TT(C)|}$. We calculate the percentage as the average of $PRST(C_{orig}, C_{par})$ for all the pairs (C_{orig}, C_{par}). As an example, if $|TT(C_{orig})| = 4$ and only 3 matching terms were found in C_{par}, the scientific term preserved (%) would be calculated as 0.75.

As indicated in Table 2 and Table 3, T5-ParEvo preserves all technical terms, as expected given that claims are filtered for such preservation. Each of the related works has substantially lower preservation values, suggesting that many scientific terms were modified or replaced. Given the rarity of correct synonyms for medical terms (i.e., drug names, disease names, enzyme names, etc don't often have synonyms), this result suggests that many generated claims will no longer be semantically equivalent in the scientific context.

4.2.3 Measure 3: Two Way Entailment Preserved (%). We used our entailment filter (described in Section 3.2) to determine the percentage of pairs (C_{orig}, C_{par}) passing this component of the Semantic Checker. From Section 3.2, we recall that given the original claim C_{orig} and the paraphrased attack claim C_{par}, to ensure that the semantics of the two claims is identical we impose a two-way entailment, i.e., $C_{orig} \models C_{par} \wedge C_{par} \models C_{orig}$. Please note that if we consider only $C_{orig} \models C_{par}$ (or alternatively $C_{par} \models C_{orig}$) passing the test, then C_{orig} (C_{par}) can be a more general claim than C_{par} (C_{orig}). Two-way entailment more strongly enforces the semantic equivalence. Again in Table 2 and Table 3, T5-ParEvo perfectly preserves entailment, as claims are filtered upon this metric. CheckList and TextFooler show high performance in this measure (though lower than T5-ParEvo), while BAE and Bert Attack score lower. Less than 40% of the claims generated by BAE preserved this measure of logic.

4.2.4 Measure 4: N-gram Manhattan Distance. We propose a measure called N-gram Manhattan Distance distance measure the syntactic differences between the original and attack claims. Given a pair (C_{orig}, C_{par}), we compute the set of all n-grams for each claim. We then calculate a vectorial representation - for each claim in the pair - with one component for each N-gram in the set, where the value of the component is the number of times that the N-gram appears in the text. Finally, we compute the Manhattan distance between the vectorial representations of C_{orig} and C_{par}). Larger distances indicate a greater difference between C_{orig} and C_{par}) in terms of their structure. Importantly, two sentences with different structures can still have the same semantics, as shown in the introductory example. Herein lies the main benefit of T5ParEvo, i.e., it creates paraphrased claims attacks that are diverse and different from the original, but semantically equivalent to it. This is optimal for our stated purpose, namely, to maximally test vulnerabilities of scientific claim verification tools. Table 2 and Table 3 show that for uni, bi, and -tri grams Manhattan Distance, T5-ParEvo has approximately twice the distance between the

[10]https://docs.scipy.org/doc/scipy-0.18.1/reference/generated/scipy.optimize.linear_sum_assignment.html

original claim and generated claim in comparison with the related works. This, with the results of the Semantic Checker, represents the tendency of our model, while still preserving the semantics (see Section 4.2.1), to modify entire phrases and restructure sentences, rather than simply exchange a single synonym as was observed with the other models. This is confirmed via the qualitative analysis in Section 4.3.

4.2.5 Measure 5: Average number of Grammatical Errors. We use *language_tool_python*[11] to determine the number of grammatical errors in each generated claim. Table 2 and Table 3 show that the average number of grammatical errors per T-5ParEvo attack claim is less than half that for BAE and Bert Attack, and considerably lower than TextFooler and CheckList. This is also confirmed by Section 4.3) where a simple exchange of synonyms was most often observed as the attack strategy for all models except T5-ParEvo. Importantly, the synonyms used for replacement did not consider the context of the sentence, including modifiers, tense, or plurality, which often led to the generation of grammatically incorrect sentences.

4.2.6 Measures 6: Readability. To ensure that successful diversified attacks generated by T5-ParEvo do not result from generation of statements too complex to be reasonably understood by the VERISCI model, we measured the readability scores of the attacks generated by each attack model. Table 4 shows results from four indices of readability. Each scores readability in terms of the level of education expected to reliably understand the statement. For the Flesch Reading Ease (FRE) score [8], a value between 10 and 30 is expected to be readable for graduate students, while a score between 30 and 50 is readable for undergraduates. For Flesch-Kincaid Grade Level(FKGL) the value represents the actual grade level expected to understand the statement [19]. For FKGL, a value of 12 indicates that a high school senior should find the statement readable. For both FRE and FKGL, the readability is measured in terms of words per sentence, and syllables per word. The Coleman-Liau Index (CLI) represents a direct grade level as well, but CLI uses a different metric to determine the readability. Where the other indices use syllables to determine readability, CLI uses number of characters [5]. The Gunning-Fog Index (GFI) represents the years of education a person is required to understand a given text, using similar metrics as FRE and FKGL [12]. By all indices, T5-ParEvo generates claims that are more readable than the original SCIFACT dataset. In fact, attacks from all models are similar in readability to the original claims. Given that the related works tended to simply exchange synonyms to generate attacks, it is unsurprising that their readability scores do not differ from the original. However, we show in Section 4.3 that T5-ParEvo generates far more sophisticated attacks, without impacting readability. This can be also seen by analyzing the results in Table 2 and Table 3 for N-gram Manhattan Distance (see Section 4.2.4). We conclude that successful diversified attacks were accomplished while keeping the same level of complexity as those of the related works. By all quantitative measures, T5-ParEvo generates superior attack claims via the rephrasing task, with respect to preserving logic, preventing grammatical errors, and maintaining the same readability levels of the related works, all while creating new claims that are far more divergent from the original. Specific examples of this are provided in Table 8 and described in detail in Section 4.3.

Table 4. Average readability scores for claims generated by each model, along with the score for the original claims in the SCIFACT dataset.

	TextFooler	Checklist	BAE	Bert Attack	T5ParEvo	Original Claims
Flesch Reading Ease (FRE)	26.58	25.10	30.07	31.18	32.73	26.70
Flesch-Kincaid Grade Level (FKGL)	12.45	12.64	11.98	11.81	11.44	11.53
Gunning-Fog Index (GFI)	15.61	15.68	15.06	15.04	14.58	15.38
Coleman-Liau Index (CLI)	18.10	18.18	17.11	17.03	16.39	17.44

[11] https://github.com/jxmorris12/language_tool_python

Table 5. Success Metrics for Attack Models Against VERISCI.

	Textfooler	Checklist	BAE	Bert Attack	T5ParEvo
(1) Successful Claims - VERISCI Attack	229	16	242	219	155
(2) Accepted Successful Claims	15	2	9	15	73
(3) % High Quality Accepted Claims	40.0	50.0	77.8	53.3	64.4
(4) % Medium Quality Accepted Claims	53.3	50.0	22.2	40.0	24.7
(5) % Low Quality Accepted Claims	6.7	0	0	6.7	11.0
(6) % Change - to/from abbreviation	0	0	0	0	0
(7) % Change - numerical to word	0	0	0	0	4.1
(8) % Change - synonym swap, nonscience word	100.0	100.0	100.0	86.7	41.1
(9) % Change - synonym swap, science word	0	0	0	0	0
(10) % Change - removed word	0	0	0	0	8.2
(11) % Change - added word	0	0	0	0	8.2
(12) % Change - sentence structure	0	0	0	0	11.0
(13) % Change - entire phrase	0	0	0	0	46.6
(14) % Change - tense	0	0	0	13.3	0
(15) % Change - multiple types	0	0	0	0	9.6
(16) % Comment - sounds odd, improper jargon	26.7	0	22.2	20.0	1.4
(17) % Comment - some grammatical errors	33.3	50.0	11.1	20.0	12.3

4.3 Qualitative Analysis

For our qualitative analysis, we enlisted a domain expert to compare the original claims with the attack claims generated against VERISCI by T5-ParEvo and the related works. This domain expert is a molecular biologist with a MS and several years' research experience in immunology, cancer, and genetics. Comparisons were made using a set of metrics by which each model could be consistently evaluated for sensibility and semantic equivalence. Descriptions of the performance of each model are categorized in terms of "success" and "acceptance". A claim deemed successful is one that reversed the decision of the target model, i.e., a successful attack. Successful attacks were then given a second level of review by the domain expert, and "accepted" claims are those which were: (i) successful attacks, (ii) grammatically correct, and (iii) preserved the scientific meaning of the original claim. Tables 5 and 6 both depict the number of successful attacks (Line 1) for each attack model tested. The remainder of Table 5 describes measured characteristics for accepted successful claims, while Table 6 describes rejection characteristics.

Table 5 Line 1 shows the number of successful attack claims for each model. As previously described, T5-ParEvo produces a number of attack claims for each original claim, while the related works return only a single attack for each. As a result, for T5-ParEvo we report the number of original claims for which *at least one* generated claim is a successful attack. This means that there may be multiple successful attacks produced from our approach, but only one will be counted to maintain consistency with the related works. At first glance, Textfooler, BAE, and Bert Attack produced a large number of claims that were able to reverse the decision of the scientific claim verification tool, while T5-ParEvo produced somewhat fewer. Checklist was markedly less able to attack the SCV tool, with only 16 successful attacks. Further scrutiny from a domain expert, however, reveals that only a small portion of those successful attacks were valid, when sensibility and preservation of scientific meaning were required. Table 5 Line 2 shows that for each of the related works, less than 15% of these successful attack claims were accepted as valid, compared to more than half for T5-ParEvo. This favors T5-ParEvo as a model with

Table 6. Rejection Metrics for Attack Models Against VERISCI.

	Textfooler	Checklist	BAE	Bert Attack	T5ParEvo
(1) Successful Claims - VERISCI Attack	229	16	242	219	155
(2) Rejected Successful Claims	214	14	232	204	82
(3) % High Quality Rejected Claim	59.7	7.1	36.0	6.0	30.6
(4) % Med Quality Rejected Claim	34.3	71.4	40.0	34	50.0
(5) % Low Quality Rejected Claim	7.1	21.4	24.0	60	19.4
(6) % Science Based Rejection	88.2	92.9	84.0	52.0	66.7
(7) % Grammar Rejection	12.0	7.1	16.0	48.0	33.3
(8) % Change - synonym swap non-science word	70.6	50.0	64.0	46.0	25.0
(9) % Change - synonym swap science word	23.5	50.0	32.0	28.0	2.8
(10) % Change - sentence restructure	0	0	0	0	33.3
(11) % Change - removed words	0	0	0	0	13.9
(12) % Change - added words	0	0	0	0	11.1
(13) % Change - numerical to word	0	0	0	0	5.6
(14) % Change - entire phrase	0	0	0	0	36.1
(15) % Change - multiple synonyms	5.9	0	12.0	28.0	11.1
(16) % Error - nonsensical statement grammar	12.0	0	20.0	44.0	22.2
(17) % Error - nonsensical statement fact or science	64.7	57.1	44.0	24.0	22.2
(19) % Error - changed meaning	23.5	35.7	36.0	8.0	47.2
(20) % Error - directly reversed meaning	0	7.1	0	28.0	0

capability to analyze robustness and identify true weaknesses of a scientific claim verification tool, rather than simply create nonsense or invalid claims which do not represent meaningful attacks. We now report specific characteristics of the accepted claims for each model.

4.3.1 Accepted Quality Grade, Table 5 Lines 3-5. We rank the accepted claims as high, medium, and low quality, where a high quality claim is one that shows no compromise in grammar and scientific meaning, and a low quality claim either has slight grammar mistakes or the meaning is less well-preserved compared to the original claim. For BAE and T5-ParEvo, a majority of claims were of high quality, while the other related works hovered around half or lower. Table 7 Lines 1-5 give an example high quality accepted attack claim from each model, along with lower quality accepted claims for each (Lines 6-14).

4.3.2 Accepted Attack Strategy, Table 5, Lines 6-15. Here we describe possible strategies used by each of the attack models, including synonym exchange, (either scientific term or not), removal or addition of words, sentence restructuring, phrase changes, and tense changes (i.e., past to present or vice versa). Notably, only T5-ParEvo attempted *any* strategy other than synonym exchange. The only exception to this was a couple of change-of-tense seen with the Bert Attack Model. Interestingly, these were actually synonym exchanges to a tense-modified word (rather than a comprehensive change-of-tense) that was technically correct. BAE is stated to have the ability to add and remove words; we observed no successful attack claims for which this was the case.

Table 7. Examples of Accepted Claims for each Attack Model. Red text indicates changes from the original claim to the attack claim, if the change is a word exchange. Note, sentence restructuring is not indicated by color.

Acceptance Type	T5ParEvo / Related Works	Original Claim	Attack Claim
Accepted - High Quality	T5ParEvo	(1) ALDH1 expression is associated with better breast cancer outcomes.	Expression of ALDH1 is associated with improved breast cancer outcomes.
	Bert-Attack	(2) Female carriers of the Apolipoprotein E4 (APOE4) allele have decreased risk for dementia.	Female carriers of the Apolipoprotein E4 (APOE4) allele have reduced risk for dementia.
	TextFooler	(3) High levels of CRP reduces the risk of exacerbations in chronic obstructive pulmonary disease (COPD).	Elevated levels of CRP reduces the risk of exacerbations in chronic obstructive pulmonary disease (COPD).
	CheckList	(4) A diminished ovarian reserve is a very strong indicator of infertility, even in an a priori non-infertile population.	A low ovarian reserve is a very strong indicator of infertility, even in an a priori non-infertile population.
	BAE	(5) Upon viral challenge, influenza-specific memory CD4+ T cells greatly enhance the early production of inflammatory chemokines in the lung.	Upon viral challenge, influenza-specific memory CD4+ T cells effectively enhance the early production of inflammatory chemokines in the lung.
Accepted - improper jargon	T5ParEvo	(6) The risk of cancer rises with level of alcohol consumption.	The risk of cancer rises with alcohol drinking level.
	Bert-Attack	(7) Cost effectiveness evaluations based on cRCT data lack external validity.	Cost effectiveness evaluations performed on cRCT data lack external validity.
	TextFooler	(8) A total of 1,000 people in the UK are asymptomatic carriers of vCJD infection.	A total of 1,000 countrymen in the UK are asymptomatic carriers of vCJD infection.
	BAE	(9) Scapular stabilizer exercises are more effective than general exercise therapy in reducing pain and improving function of the shoulder.	Scapular stabilizer exercises are more effective than systemic exercise therapy in reducing pain and improving function of the shoulder.
Accepted - grammatical errors	T5ParEvo	(10) Vitamin D is an important factor in the relationship between calcium and parathyroid hormone.	Vitamin D is a major factor in the relationship between calcium and parathyroid hormone
	Bert-Attack	(11) Chlamydia trachomatis is most prevalent in the UK among sexually-experienced individuals aged 16 to 24.	Chlamydia trachomatis is most prevalent in the britain among sexually-experienced individuals aged 16 to 24.
	TextFooler	(12) 76-85% of people with severe mental disorder receive no treatment in low and middle income countries.	76-85% of people with severe mental disorder receiving no treatment in low and middle income countries.
	CheckList	(13) 76-85% of people with severe mental disorder receive no treatment in low and middle income countries.	76-85% of people with severe mental disorder receiving no treatment in low and middle income countries.
	BAE	(14) High levels of CRP reduces the risk of exacerbations in chronic obstructive pulmonary disease (COPD).	elevated levels of CRP reduces the risk of exacerbations in chronic obstructive pulmonary disease (COPD).

4.3.3 Acceptance Comments, Table 5 Line 16-17. Our qualitative analysis revealed a large number of attack claims for the related works which were accepted but low quality due to two very common issues: improper jargon and some grammatical errors. Improper jargon refers to words which are technically correct, but certainly not the term-of-art used in scientific statements and thus sound odd to the trained ear. Additionally, a large number of accepted claims showed small grammatical errors. Table 7 shows examples of high quality accepted claims along with examples of each of these common error types.

When considering the types of successful modifications to the original claim, T5-ParEvo demonstrates markedly increased diversity in strategy for attack generation compared to the related works (Table 5, Lines 6-15). In fact, the attack methodology for Textfooler, BAE, and CheckList is perfectly homogeneous: exchange of important words. For acceptable claims, these were non-science words, as replacing scientific terms results in changed

Table 8. Examples of Diverse Attack Claims Generated by T5ParEvo. Depicted are representative examples of accepted claims which used strategies not employed by any of the related works.

Change	Original Claim	Attack Claim
Added Words	(1) 76-85% of people with severe mental disorder receive no treatment in low and middle income countries.	76-85 percent of people with severe mental disorder will receive no treatment in low and middle income countries.
	(2) Risk-adjusted mortality rates are similar in teaching and non-teaching hospitals.	Similar risk-adjusted mortality rates in teaching and non-teaching hospitals are reported.
Phrase Changes	(3) ALDH1 expression is associated with better breast cancer outcomes.	The expression of ALDH1 is related to better breast cancer outcomes.
	(4) Antimicrobial agents are less effective due to the pressure of antimicrobial usage.	Antimicrobials are less effective due to pressure from antimicrobial use.
	(5) ART has no effect on the infectiveness of HIV-positive people.	ART did not change the infectiveness of HIV-positive people.
	(6) Bariatric surgery has a positive impact on mental health.	Bariatric surgery has measurable psychological benefits.
	(7) Dexamethasone decreases risk of postoperative bleeding.	Dexamethasone decreases postoperative bleeding risk.
	(8) The risk of female prisoners harming themselves is ten times that of male prisoners.	The risk of female prisoners harming themselves is 10 times greater than male prisoners.
	(9) Stroke patients with prior use of direct oral anticoagulants have a lower risk of in-hospital mortality than stroke patients with prior use of warfarin.	Stroke patients with prior use of direct oral anticoagulants have a lower mortality rate in-hospital than stroke victims who had used warfarin previously.
Generalization	(10) Antimicrobial agents are more effective due to the pressure of antimicrobial usage.	Antimicrobial agents are due to their pressure more effective.
Sentence Restructuring	(11) 76-85% of people with severe mental disorder receive no treatment in low and middle income countries.	76-85% of people with severe mental disorder in low and middle income countries receive no treatment.
	(12) Anthrax spores can be disposed of easily after they are dispersed.	Anthrax spores can be easily disposed of once they are dispersed.
	(13) Gene expression does not vary appreciably across genetically identical cells.	Gene expression does not differ across genetically identical cells appreciably.
	(14) Incidence of 10/66 dementia is lower than the incidence of DSM-IV dementia.	The prevalence of DSM-IV dementia is higher than the incidenceof 10/66 dementia.
	(15) Incidence of sepsis has fallen substantially from 2009 to 2014.	From 2009 to 2014 the prevalence of sepsis has fallen considerably.

meaning for nearly all scientific claims. This reveals T5-ParEvo as a superior attack model through which we can identify (and potentially mitigate via further training) vulnerabilities of scientific claim verification tools.

Table 6 shows the rejection metrics collected for each model. Again, Lines 1 and 2 demonstrate that, though many of the related works appeared to be more successful in attacking VERISCI, the vast majority of these attacks are not valid. Below, we detail the rejection metrics used to analyze the attack claims for T5-ParEvo and its related works.

4.3.4 Rejected Quality, Table 6 Lines 3-5. As with Accepted Claims, we scored each rejected claim for quality (high, medium, or low) based upon its change of meaning and sensibility. TextFooler, BAE, and T5-ParEvo had the greatest number of higher quality rejected claims, while Bert Attack had a notably high percentage of low

quality rejections. Importantly, these represent those claims which were either completely unreadable nonsense, or those which directly reversed the scientific meaning of the original claim. This was a significant problem with Bert Attack: it tended to replace words with antonyms rather than synonyms. These clearly do not represent valid attacks, as the SCV tool is correct in reversing its classification when the claim is also reversed in meaning.

4.3.5 Rejection Basis, Table 6 Lines 6-7. We decompose the rejections into those rejected based upon science (or fact) or those based upon grammar. Science-based rejections refer to those which change the meaning, or are nonsensical with respect to science or fact. We provide Table 10 in Appendix X which gives examples of claims rejected based upon science along with rejections based upon grammar. For all models, the largest proportion of rejections were on the basis of incorrect science or fact. Only Bert Attack had a lower percentage of science or fact rejections than T5-ParEvo, largely due to its tendency to generate grammatical nonsense (Table 6 Line 16).

4.3.6 Rejected Strategy, Table 6 Lines 8-15. Here we describe again the strategies used to attack VERISCI for T5-ParEvo and each related work. Again we observe that only our approach attempts a diverse set of strategies to fool the SCV tool. Notably, comparing Table 6 Line 15 with Table 5 Line 15 shows that most of the models attempted to exchange more than one synonym, but only T5-ParEvo was able to do so in a way that preserved sensibility and scientific meaning.

4.3.7 Errors, Table 6 Lines 16-20. Above, we described the rejected attacks in terms of the attack strategies. Here, we describe the actual errors that led to rejection by the domain expert. We calculated the percent of rejected attacks that were nonsensical statements, then further divided those into nonsense science or nonsense grammar. Table 10 in Appendix X shows examples of each. As a single example, for the original claim - "Antidepressants increase the severity of migraines." Checklist returned a successful claim attack - "adults increase the severity of migraines." Though this attack is grammatically correct, it is clearly nonsense with respect to science. Other errors measured included changed meaning, or direct reversal of meaning, which we distinguished to show the antonym exchange occurring with Bert Attack, and to some degree, CheckList.

None of the related works demonstrated a satisfactory level of complexity in their attacks, generating statements that on first glance look like writing observed when one indiscriminately uses a thesaurus. On occasion, swapping in a word randomly will be successful; most times it will not. In contrast, the T5-ParEvo models generate claims via paraphrasing, which results in a large diversity of candidate attacks. Table 8 shows a few examples of the strategies employed: sentence restructuring (Claim 11-15), adding words (Claims 1 and 2), removing unnecessary words or generalizing (Claim 10), changing entire phrases (Claim 3-9), and even change of voice (Claim 2) were employed. Numbers and symbols were replaced with their spelled words or vice versa (Claims 1 and 8). Further, multiple strategies were often utilized within a single attack claim, i.e. change of terms or phrases along with sentence restructuring (Claim 9, 11-15). T5-ParEvo models also generate multiple candidate claims where possible, again yielding a greater variety of more complex claims. Not only were modifiers appropriately altered, but a greater understanding of the nuance of language was demonstrated. For instance, the model recognizes that, when a subject is referred to twice in the sentence, a general pronoun can replace the second reference (Claim 10). As another example the model realized that "ten times that of…" meant "10 times greater than…." (Claim 8). Changing entire phrases also prevents the problem of non-matching modifiers, which tended to generate grammatical nonsense.

4.4 Distinguishing Between Machine Generated Adversarial Attacks and Human-Generated Claims

In this section, we analyze whether it is possible to create a classification model able to distinguish human-generated claims from machine-generated adversarial attacks. To do so, for each attack model, we created a dataset with real human-generated claims and the generated adversarial attacks. Given each dataset associated with a specific attack model, we performed a stratified 10-fold cross-validation to fine-tune and test a classification

model based on MPNET [43]. MPNET is state of art neural network already pretrained that, when fine-tuned in specific text classification tasks, through transfer learning, usually provides results comparable to the best state of the art. For each fold, the test set is oversampled to produce a same-size test set for each attack model. This produces comparable accuracy scores among the different tasks. Each accuracy score measures the possibility of distinguishing Machine Generated Adversarial Attacks for each attack model and real human-generated claims.

Table 9. Accuracy results for distinguishing Machine Generated Adversarial Attacks from real human-generated claims

Textfooler	Checklist	BAE	Bert Attack	T5ParEvo
0.44	0.49	0.42	0.46	0.52

Table 9 reports the accuracy score for each attack model. As can be seen in the table, none of the generated adversarial attacks is distinguishable from the real human generated claims through MPNET. The best classification result in Table 9 is equivalent to equivalent to a coin flip. This shows the ability of all the attack models including T5-ParEvo to create adversarial examples not easily detectable as generated. Each of the models we tested, including our own, generate attack claims that a pretrained classifier is unable to distinguish from human-generated claims. As with the readability study, we use this analysis to demonstrate that observed diversity and successful attacks from T5-ParEvo are not due to generation of claims that are exceedingly complex or unnatural compared to the original (written by human).

4.5 Effectiveness of Fine-Tuning the T5 Model and the Entailment Module

As described in the methodology (Section 3.2), the Natural Language Inference model (roberta.large .mnli) responsible for detecting entailment was fine tuned using the SciTail Dataset to achieve optimal results in the scientific domain. We compared the number of unique claims generated with the T5-ParEvo model with and without a fine-tuned NLI and T5 model as an ablation study. Without fine tuning, only 522 successful candidate claims were generated, representing 14 unique original claims. For T5-ParEvo including a fine-tuned NLI for entailment semantic checking and the T5 model with biological datasets, over 3600 successful candidate claims were produced, representing 155 unique original claims. Considering this, it is apparent that fine-tuning of the entailment model for the specific scientific domain is an important step for successful paraphrasing attacks.

4.6 Effectiveness of T5-ParEvo Iterations

T5-ParEvo is an evolutionary procedure, in which the pretrained T5 model generates paraphrased claims, followed by attack of the SCV tool using those claims. At each iteration, successful attacks from the previous iteration are used to further fine-tune the model. Analyzing Figure 3 it is observed that at each iteration, the number of unique original claims successfully attacked increases. This demonstrates that at each iteration, the model extends learned attack strategies to other claims. Moreover, since the number of unique pairs also increases, it follows that T5-ParEvo continually finds new ways to attack the original claims.

This reveals the relevancy of such an evolutionary procedure. In addition, as an ablation study, we generated with the pretrained T5, 400 paraphrased claims for each original claim, and selected only those that passed the semantic checker and reverted the classification. The result (the number of unique claims and unique pairs) is similar to the result reported in Figure 3 for iteration 0, i.e., the results without fine-tuning T5. From this result it is clear that the T5-ParEvo procedure improves the T5 original ability to generate not only successful, but diverse attacks.

5 CONCLUSION AND FUTURE WORKS

SCV tools have great potential for real-world use, but suffer from an important weakness: claims that are identical in scientific meaning but stated differently are often classified differently. Adversarial attack models can assist in

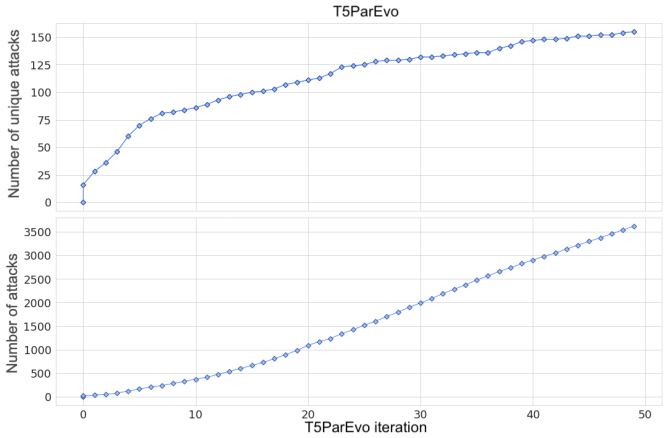

Fig. 3. Performance of **T5 – ParEvo** at each iteration

improving these tools by determining weaknesses and suggesting avenues of further training. In this work we propose a paraphrasing model-based attack to SCV tools, T5-ParEvo, which operates via a different strategy than current attack models. Rather than simply deducing important words in a statement, then attempting to change those words, the entire statement is considered. T5-ParEvo, compared to other models, produced a substantially greater number of attacks against the SVC tools that preserved the scientific meaning of the original statement. Other models produced a majority of claims that reversed the result of the SCV, but did so because the actual meaning of the statement had been changed, in some cases completely reversed. Multiple different strategies to modify claims are employed by T5-ParEvo, and potentially a variety of diverse candidate statements are produced for each claim. A greater diversity of attacks are likely to more comprehensively test an SCV tool for its weaknesses. For example, we have shown that a simple restructuring of a statement can cause SCV tools such as VERISCI, MultiVerS, ARSJoint, and ChatGPT to reverse their stance about a scientific claim, even when the words of the statement are unchanged. Importantly, this weakness could not be revealed by the existing attack models. Future work will examine the potential for T5-ParEvo to enhance training sets for SCV tools by generating the same claims in a variety of forms.

6 ACKNOWLEDGEMENT

This work was supported in part by the Office of Naval Research grant N00014-18-1-2670 and N00014-23-1-2132, and by the U.S. National Science Foundation grant CNS-1822094.

REFERENCES

[1] A. Aizawa. 2003. An information-theoretic perspective of tf–idf measures. *Information Processing & Management* 39, 1 (2003), 45–65.
[2] M. Alzantot, Y. Sharma, A. Elgohary, B.J. Ho, M. Srivastava, and K.W. Chang. 2018. Generating natural language adversarial examples. *arXiv preprint arXiv:1804.07998* (2018).
[3] I. Beltagy, M. E. Peters, and A. Cohan. 2020. Longformer: The Long-Document Transformer. *arXiv:2004.05150* (2020).
[4] S. R. Bowman, G. Angeli, C. Potts, and C. D. Manning. 2015. A large annotated corpus for learning natural language inference. *arXiv preprint arXiv:1508.05326* (2015).
[5] M. Coleman and T. L. Liau. 1975. A computer readability formula designed for machine scoring. *Journal of Applied Psychology* 60, 2 (1975), 283.
[6] J. DeYoung, E. Lehman, B. Nye, I. J. Marshall, and B. C. Wallace. 2020. Evidence Inference 2.0: More Data, Better Models. arXiv:2005.04177 [cs.CL]

[7] J. Ebrahimi, A. Rao, D. Lowd, and D. Dou. 2017. Hotflip: White-box adversarial examples for text classification. *arXiv preprint arXiv:1712.06751* (2017).
[8] R. Flesch. 1948. A new readability yardstick. *Journal of applied psychology* 32, 3 (1948), 221.
[9] J. Gao, J. Lanchantin, M.L. Soffa, and Y. Qi. 2018. Black-box generation of adversarial text sequences to evade deep learning classifiers. In *2018 IEEE Security and Privacy Workshops (SPW)*. IEEE, 50–56.
[10] S. Garg and G. Ramakrishnan. 2020. Bae: Bert-based adversarial examples for text classification. *arXiv preprint arXiv:2004.01970* (2020).
[11] M. Glockner, V. Shwartz, and Y. Goldberg. 2018. Breaking NLI systems with sentences that require simple lexical inferences. *arXiv preprint arXiv:1805.02266* (2018).
[12] R. Gunning et al. 1952. Technique of clear writing. (1952).
[13] M. Iyyer, J. Wieting, K. Gimpel, and L. Zettlemoyer. 2018. Adversarial example generation with syntactically controlled paraphrase networks. *arXiv preprint arXiv:1804.06059* (2018).
[14] E. Jang, S. Gu, and B. Poole. 2016. Categorical reparameterization with gumbel-softmax. *arXiv preprint arXiv:1611.01144* (2016).
[15] R. Jia and P. Liang. 2017. Adversarial examples for evaluating reading comprehension systems. *arXiv preprint arXiv:1707.07328* (2017).
[16] D. Jin, Z. Jin, J T Zhou, and P. Szolovits. 2020. Is BERT Really Robust? A Strong Baseline for Natural Language Attack on Text Classification and Entailment. In *Proceedings of the AAAI conference on artificial intelligence*, Vol. 34. 8018–8025.
[17] Q. Jin, B. Dhingra, Z. Liu, W. Cohen, and X. Lu. 2019. PubMedQA: A Dataset for Biomedical Research Question Answering. 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)*. 2567–2577.
[18] T. Khot, A. Sabharwal, and P. Clark. 2018. Scitail: A textual entailment dataset from science question answering. In *Thirty-Second AAAI Conference on Artificial Intelligence*.
[19] JP Kincaid, RP Fishburne, RL Rogers, and BS Chissom. 1975. Research branch report 8–75. *Memphis: Naval Air Station* (1975).
[20] H. Kwon and S. Lee. 2022. Ensemble transfer attack targeting text classification systems. *Computers & Security* 117 (2022), 102695. https://doi.org/10.1016/j.cose.2022.102695
[21] Z. Lan, M. Chen, S. Goodman, K. Gimpel, P. Sharma, and R. Soricut. 2019. Albert: A lite bert for self-supervised learning of language representations. *arXiv preprint arXiv:1909.11942* (2019).
[22] Yoon Wonjin Kim Sungdong Kim Donghyeon Kim Sunkyu Ho Chan So Kang Jaewoo Lee, Jinhyuk. 2020. Biobert: a pre-trained biomedical language representation model for biomedical text mining. *Bioinformatics* 36 (2020), 1234–1240.
[23] E. Lehman, J. DeYoung, R. Barzilay, and B C Wallace. 2019. Inferring Which Medical Treatments Work from Reports of Clinical Trials. In *Proceedings of the North American Chapter of the Association for Computational Linguistics (NAACL)*. 3705–3717.
[24] J. Li, S. Ji, T. Du, B. Li, and T. Wang. 2018. Textbugger: Generating adversarial text against real-world applications. *arXiv preprint arXiv:1812.05271* (2018).
[25] L. Li, R. Ma, Q. Guo, X. Xue, and X. Qiu. 2020. BERT-ATTACK: Adversarial Attack Against BERT Using BERT. *CoRR* abs/2004.09984 (2020). arXiv:2004.09984 https://arxiv.org/abs/2004.09984
[26] X. Li, G A Burns, and N. Peng. 2021. A Paragraph-level Multi-task Learning Model for Scientific Fact-Verification.. In *SDU@ AAAI*.
[27] B. Liang, H. Li, M. Su, P. Bian, X. Li, and W. Shi. 2017. Deep text classification can be fooled. *arXiv preprint arXiv:1704.08006* (2017).
[28] A. Liu, H. Yu, X. Hu, S. Li, L. Lin, F. Ma, Y. Yang, and L. Wen. 2022. Character-level White-Box Adversarial Attacks against Transformers via Attachable Subwords Substitution. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 7664–7676. https://aclanthology.org/2022.emnlp-main.522
[29] H. Liu, Z. Xu, X. Zhang, X. Xu, F. Zhang, F. Ma, H. Chen, H. Yu, and X. Zhang. 2023. SSPAttack: A Simple and Sweet Paradigm for Black-Box Hard-Label Textual Adversarial Attack. *Proceedings of the AAAI Conference on Artificial Intelligence* 37, 11 (2023), 13228–13235. https://doi.org/10.1609/aaai.v37i11.26553
[30] J. Liu, H. Jin, G. Xu, M. Lin, T. Wu, M. Nour, F. Alenezi, A. Alhudhaif, and K. Polat. 2023. Aliasing black box adversarial attack with joint self-attention distribution and confidence probability. *Expert Systems with Applications* 214 (2023), 119110. https://doi.org/10.1016/j.eswa.2022.119110
[31] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, and V. Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692* (2019).
[32] D. Nadeau and S. Sekine. 2007. A survey of named entity recognition and classification. *Lingvisticae Investigationes* 30, 1 (2007), 3–26.
[33] W. Nie, N. Narodytska, and A. Patel. 2018. Relgan: Relational generative adversarial networks for text generation. In *International conference on learning representations*.
[34] OpenAI. 2023. ChatGPT. https://chat.openai.com/.
[35] A. Pétrowski and S. Ben-Hamida. 2017. *Evolutionary algorithms*. John Wiley & Sons.
[36] R. Pradeep, X. Ma, R. Nogueira, and J. Lin. 2020. Scientific claim verification with VerT5erini. *arXiv preprint arXiv:2010.11930* (2020).
[37] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, I. Sutskever, et al. 2019. Language models are unsupervised multitask learners. *OpenAI blog* 1, 8 (2019), 9.

[38] C. Raffel, N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Zhou, W. Li, and P J Liu. 2019. Exploring the limits of transfer learning with a unified text-to-text transformer. *arXiv preprint arXiv:1910.10683* (2019).

[39] R Ramachandran and K Arutchelvan. 2021. Named entity recognition on bio-medical literature documents using hybrid based approach. *Journal of Ambient Intelligence and Humanized Computing* (2021), 1–10.

[40] S. Ren, Y. Deng, K. He, and W. Che. 2019. Generating natural language adversarial examples through probability weighted word saliency. In *Proceedings of the 57th annual meeting of the association for computational linguistics*. 1085–1097.

[41] M.T. Ribeiro, T. Wu, C. Guestrin, and S. Singh. 2020. Beyond Accuracy: Behavioral Testing of NLP models with CheckList. In *Association for Computational Linguistics (ACL)*.

[42] S. Sadrizadeh, L. Dolamic, and P. Frossard. 2022. Block-Sparse Adversarial Attack to Fool Transformer-Based Text Classifiers. In *ICASSP 2022 - 2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. 7837–7841. https://doi.org/10.1109/ICASSP43922.2022.9747475

[43] K. Song, X. Tan, T. Qin, J. Lu, and T.Y. Liu. 2020. Mpnet: Masked and permuted pre-training for language understanding. *Advances in Neural Information Processing Systems* 33 (2020), 16857–16867.

[44] J. Thorne, A. Vlachos, C. Christodoulopoulos, and A. Mittal. 2018. FEVER: a Large-scale Dataset for Fact Extraction and VERification. In *NAACL-HLT*.

[45] Petter Törnberg. 2023. Chatgpt-4 outperforms experts and crowd workers in annotating political twitter messages with zero-shot learning. *arXiv preprint arXiv:2304.06588* (2023).

[46] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A N Gomez, Ł Kaiser, and I. Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems* 30 (2017).

[47] D. Wadden, S. Lin, K. Lo, L L Wang, M. van Zuylen, A. Cohan, and H. Hajishirzi. 2020. Fact or Fiction: Verifying Scientific Claims. In *Proc. of EMNLP*. Association for Computational Linguistics, Online, 7534–7550. https://doi.org/10.18653/v1/2020.emnlp-main.609

[48] D. Wadden, K. Lo, LL Wang, A. Cohan, I. Beltagy, and H. Hajishirzi. 2021. LongChecker: Improving scientific claim verification by modeling full-abstract context. *arXiv preprint arXiv:2112.01640* (2021).

[49] D. Wadden, K. Lo, L. Wang, A. Cohan, I. Beltagy, and H. Hajishirzi. 2022. MultiVerS: Improving scientific claim verification with weak supervision and full-document context. In *Findings of the Association for Computational Linguistics: NAACL 2022*. 61–76.

[50] D. Wadden, K. Lo, L. Lu Wang, A. Cohan, I. Beltagy, and H. Hajishirzi. 2021. LongChecker: Improving scientific claim verification by modeling full-abstract context. *CoRR* abs/2112.01640 (2021). arXiv:2112.01640 https://arxiv.org/abs/2112.01640

[51] A. Williams, N. Nangia, and S. R. Bowman. 2017. A broad-coverage challenge corpus for sentence understanding through inference. *arXiv preprint arXiv:1704.05426* (2017).

[52] Y. Xian, B. Schiele, and Z. Akata. 2017. Zero-shot learning-the good, the bad and the ugly. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 4582–4591.

[53] Y. Zhang, J. Baldridge, and L. He. 2019. PAWS: Paraphrase Adversaries from Word Scrambling. In *Proc. of NAACL*.

[54] Z. Zhang, J. Li, F. Fukumoto, and Y. Ye. 2021. Abstract, Rationale, Stance: A Joint Model for Scientific Claim Verification. *arXiv preprint arXiv:2110.15116* (2021).

A SUPPLEMENTAL QUALITATIVE ANALYSIS

Tables 10 and 11 depict example rejected attacks from T5-ParEvo and each of the related works. These examples showcase the types of errors made by each model, evaluated in the Qualitative Analysis.

B ITERATIVE FINE-TUNING ON MULTIVERS AND ARSJOINT

We previously showed that the iterative fine-tuning of T5-ParEvo improved its ability against VERISCI to produce more successful and diverse attack claims for each original. Here, we confirm that this is also true for both MultiVerS and ARSJoint, other state-of-the-art SCV tools. It is clear from Figure 4 that with increasing iterations, a larger number of original claims could be successfully attacked for both SCV tools. We also observed increase number of total attacks (not shown), indicating that the diversity of attack strategy was iteratively improved using our approach.

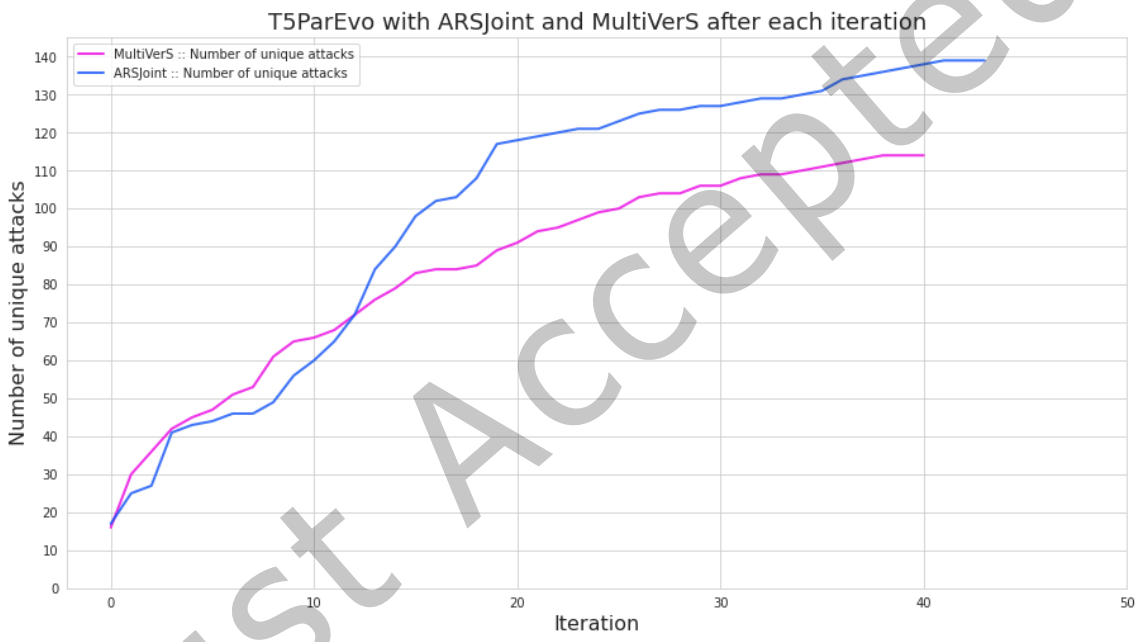

Fig. 4. Performance of **T5 – ParEvo** at each iteration

C PARAPHRASING MODELS

We compare several paraphrasers before settling upon T5 for incorporation into our model. The paraphrasing model that best suits our task will: (i) generate a large number of paraphrased claims for each original, (ii) have a high percentage of claims which preserve scientific terms and two-way entailment, (iii) generate claims syntactically different from the original, and (iv) have a low number of grammatical errors. This is detailed in our Experimental Section (4.2) A large number of paraphrased claims ensures diversity of attacks. A high percentage of paraphrased claims that preserve scientific terms and entailment allows us to have a large initial set of claims to begin the iterative fine-tuning procedure of T5-ParEvo.

Table 12 compares the T5base[12], T5PAWS[13] (our base paraphraser, described in main paper Section 3.1), and the two next best performing paraphrasers, bart_{eugene}[14] and ProtAugment[15]. When we comprehensively consider the characteristics of each, we see that both bart_{eugene} and ProtAugment have a higher tendency to pass the semantic checker, from the perspective of both the scientific terms preservation and entailment (described in main paper Section 3.2). However, it is also noticeable that the N-gram Manhattan distances are very low for both (Line 5), meaning that the paraphrased claims are syntactically very similar to the original claims. Additionally, both models were unable to generate an acceptable number of claims for each original (Line 1), which we required to obtain maximal diversity.

Table 13 compares four additional paraphrase models, $\text{bart}_{stanford}$[16], mt0[17], alpaca[18], and Primera[19]. Here we see that all four are able to produce a large number of paraphrases per original claim (Line 1), and they produce claims with large syntactic differences from the original (Line 5). However, each has a substantially lower percent of paraphrased claims which preserve scientific terms and entailment (Line 2), and two have a tendency to produce a large number of grammatical errors (Line 6).

We also include a sample of the paraphrased claims generated by a few models (Table 14). Similar to the related works models, $\text{bart}_{stanford}$ created many sentence fragments, or failed to use correct modifiers, creating statements with unacceptable grammar (claims 1-2). This model did attempt to restructure sentences, but was less able to do so than the T5 paraphrasing model. The bart_{eugene} model was even less capable, and created a large number of sentence fragments and truncated logic (claims 3-4). Alpaca was ambitious in its willingness to create diverse paraphrases, but often failed completely in preserving meaning (claims 5-6).

D REPRODUCIBILITY

The code for T5-ParEvo, can be found at *https://github.com/ratulalahy/T5ParEvo*, along with the claims generated by each attack model, and the individual qualitative analysis. The repository contains the following: (i) A directory for each related work that contains the code and the claims set generated for each of the models tested. (ii) A directory each for T5-ParEvo that contains the code and generated claims set (iii) A directory "IndividualAnalyses" that contains the qualitative analyses for the related works and T5-ParEvo.

E CHATGPT ATTACK

ChatGPT [34] is a state-of-the-art language model developed by OpenAI. It is trained on a diverse range of data, allowing it to generate human-like responses to various prompts. One of its remarkable features is its capacity for zero-shot learning [45, 52], enabling it to perform tasks or answer questions on topics for which it has not been explicitly trained. This showcases a level of generalization and adaptability beyond its specific training data. For this reason, we decided to investigate the capability of ChatGPT as a Scientific Claim Verification Tool and identify potential weaknesses in this task. To verify the zero-shot learning capability of ChatGPT as an SCV tool, we queried it using the following query: "Is the following claim true: <the specific claim>? Please answer *true*, *false* or *do_not_know*." We then test ChatGPT on the training set of claims in SCIFACT (no ground truth is provided for the test set) and compare the answer with the ground truth. We compute the F1-score only for those claims for which ChatGPT returns *true* or *false* (claims with *do_not_know* answers are discarded); the

[12]https://huggingface.co/t5-base
[13]https://huggingface.co/Vamsi/T5_Paraphrase_Paws
[14]https://huggingface.co/eugenesiow/bart-paraphrase
[15]https://huggingface.co/tdopierre/ProtAugment-ParaphraseGenerator
[16]https://huggingface.co/stanford-oval/paraphraser-bart-large
[17]https://huggingface.co/bigscience/mt0-base
[18]https://huggingface.co/declare-lab/flan-alpaca-large
[19]https://huggingface.co/allenai/PRIMERA-arxiv

Table 10. Examples of Rejected Claims for each Attack. Red text indicates changes from the original claim to the attack claim, if the change is a word exchange. Note, sentence restructuring is not indicated by color.

Rejection Reason	Model	Original Claim	Attack Claim
Nonsense-Grammar	T5ParEvo	(1) LDL cholesterol has no involvement in the development of cardiovascular disease. (2) Mitochondria play a trivial role in calcium homeostasis.	LDL cholesterol does no longer support in the development of cardiovascular disease. Mitochondria play trolical part in calcium homeost.
Nonsense-Grammar	Textfooler	(3) PD-1 triggering on monocytes reduces IL-10 production by monocytes. (4) Medications to treat obesity do not have side effects.	PD-1 triggering on monocytes reduces IL-10 engender by monocytes. Treatment to healing obesity do not have side effects.
Nonsense-Grammar	Bert-Attack	(5) DUSP4 decreases apoptosis. (6) Exercise increases cancer mortality rates among Chinese citizens.	.sp. decreases apoptosis. the increases cancer mortality rates among Chinese citizens.
Nonsense-Grammar	BAE	(7)Ribosomopathies have a high degree of cell and tissue specific pathology. (8) Risk of cardiovascular events can be cut by a third by using antihypertensive drug therapy among hemodialysis patients.	none have a high degree of cell and tissue specific pathology. Risk of cardiovascular events can be exceeded by a third by using antihypertensive drug therapy among hemodialysis patients.
Nonsense-Science	T5ParEvo	(9) Ly6C hi monocytes have a lower inflammatory capacity than Ly6C lo monocytes. (10) Macrolides have no protective effect against myocardial infarction.	Ly6C high monocytes have a lower inflammation capacity than those of Ly6C monocyte. Macrolides have no longer tolerability against myocardial infarction.
Nonsense-Science	Textfooler	(11) Noninvasive positive pressure ventilation is not predictive of acute respiratory failure after solid organ transplantation. (12) Whole brain radiotherapy increases the occurrence of new brain metastases.	Noninvasive useful pressure respirator is not predictive of serious respiratory failure after solid organ registers. Whole brain radiotherapy increases the apparition of new drain metastases.
Nonsense-Science	CheckList	(13) ALDH1 expression is associated with poorer prognosis for breast cancer primary tumors. (14) Antidepressants increase the severity of migraines.	ALDH1 expression is associated with poorer prognosis for breast versus primary tumors. adults increase the severity of migraines.
Nonsense-Science	Bert-Attack	(15) The most prevalent adverse events to Semaglutide are cardiovascular. (16) Angiotensin converting enzyme inhibitors are associated with increased risk for functional renal insufficiency.	The most of adverse events to Semaglutide are respiratory. Angiotensin converting enzyme inhibitors are promoted with increased risk for functional renal insufficiency.
Nonsense-Science	BAE	(17) NR5A2 is important in development of endometrial tissues. (18)Trans-acting factors, such as lncRNAs, influence mRNA translation.	inflammation is dominant in absence of endometrial tissues. certain receptors, such as adrenaline, influence mRNA translation.

Table 11. Examples of Rejected Claims for each Attack. Red text indicates changes from the original claim to the attack claim, if the change is a word exchange. Note, sentence restructuring is not indicated by color.

Changed Meaning	T5ParEvo	(19) Mice lacking Sirt1 in Sf1-expressing neurons have increased susceptibility to diet-induced obesity and insulin resistance. (20) Natriuretic peptides increase susceptibility to diabetes.	Mice lacking Sirt1 in Sf1-expressing neurons have increases in response to diet-induced obesity and insulin resistance. Natriuretic peptides increase diabetes risk factors.
Changed Meaning	Textfooler	(21) Nonsteroidal antinflammatory drugs are ineffective as cancer treatments. (22) Chenodeosycholic acid treatment reduces whole-body energy expenditure.	Nonsteroidal antinflammatory drugs are dispensable as cancer treatments. Chenodeosycholic acid treatment reduces whole-body energy burdens.
Changed Meaning	BAE	(23) The DESMOND program demonstrates no significant impact on lifestyles outcomes. (24) Having a main partner worsens HIV outcomes.	The DESMOND scale demonstrates no significant impact on school outcomes. Having a better partner worsens HIV outcomes.
Changed Meaning	Bert-Attack	(25) The DESMOND program demonstrates no significant impact on weight loss. (26) FoxO3a activation in neuronal cell death is mediated by reactive oxygen species (ROS).	The DESMOND program demonstrates no significant affect on weight development. FoxO3a activation in neuronal cell death is reduced by reactive oxygen species (ROS).
Changed Meaning	CheckList	(27) A breast cancer patient's capacity to metabolize tamoxifen influences treatment outcome. (28) Adult tissue-resident macrophages are seeded before birth. (29) Chenodeoxycholic acid treatment decreases brown adipose tissue activity.	A prostate surgery patient's capacity to metabolize tamoxifen influences treatment outcome. Adult tissue-resident macrophages are activated before birth. Chenodeoxycholic acid treatment causes brown adipose tissue activity.
Reversed Meaning	Bert-Attack	(30) Ultrasound guidance significantly deraises the number of traumatic procedures when attempting needle insertion. (31) The mean suicide rate in women is lower after miscarriage than live birth.	Ultrasound guidance significantly decreases the number of traumatic procedures when attempting needle insertion. The mean suicide rate in women is greater after miscarriage than live birth.
Reversed Meaning	CheckList	(32) Ambulatory blood pressure monitoring is inaccurate at diagnosing hypertension.	Ambulatory blood pressure monitoring is adequate at diagnosing hypertension.

zero-shot learning capability of ChaGPT achieves 0.63. This is not properly comparable with the test set F1 scores reported on the leaderboard [20] for the other SCV tools. However, these scores are nonetheless impressive when compared to those of the other SCV tools. We then tested the robustness of ChatGPT to T5-ParEvo adversarial

[20]https://leaderboard.allenai.org/scifact/submissions/public

Table 12. Quantitive analysis between T5ParEvo and other existing paraphrasing models

		T5base	T5PAWS	bart$_{eugene}$	ProtAugment
Avg. Number Paraphrased Claims per Original		3.2	8.37	1.24	1.19
% Paraphrased Claims Passing Semantic Checker		13.19	46.91	88.25	89.11
Scientific Term Preserved (%)		30.56	76.90	95.63	96.71
Two-way Entailment Preserved (%)		15.72	82.19	96.99	95.41
N-gram Manhattan Distance (mean)	Uni-gram	4.98	1.18	0.32	0.20
	Bi-gram	2.81	1.19	0.29	0.16
	Tri-gram	2.45	1.34	0.34	0.17
Avg. number of Grammatical Errors		0.28	0.86	0.65	0.73

claims. Unfortunately, ChatGPT is not accessible to attack in the manner used for other SCV tools. Therefore, we do not directly attack ChatGPT with our model, but use a hybrid attack strategy. This strategy uses VERISCI as a surrogate model for ChatGPT, and uses the attack claims generated by our T5ParEvo for VERISCI on ChatGPT. 43.87% of the successful unique attacks generated for VERISCI (68 unique attacks) are successful on ChatGPT, i.e. 43.87% of the original claims for which there exists a successful attack for VERISCI, there also exists a paraphrased claim which induces ChatGPT to produce a different outcome w.r.t. the original claim. Importantly, each of the claims tested was verified to be valid by both the quantitative and qualitative analysis. Whether or not ChatGPT has ever been trained on the SCIFACT claims, it is undeniable that paraphrasing the claims induces a change in the response of ChatGPT, representing weaknesses in its understanding and undermining human trust in it as a fact verification model. We conclude that ChatGPT is a promising generative AI model, but its robustness is an open problem that requires a large and timely consideration. It should be noted that these results are preliminary, and designed only to show that ChatGPT shares an inherent vulnerability to recognizing paraphrased sentences as semantically equivalent. More sophisticated prompting methods, such as chain-of-thought prompting, have the potential to improve ChatGPT's ability to understand paraphrased claims and therefore its success rate against T5-ParEvo attacks.

Table 13. Quantitive analysis for existing paraphrasing models

		$bart_{stanford}$	mt0	alpaca	Primera
Avg. Number Paraphrased Claims per Original		9.88	8.37	9.84	9.99
% Paraphrased Claims Passing Semantic Checker		34.95	30.24	25.07	2.53
Scientific Term Preserved (%)		65.08	70.17	72.57	45.00
Two-way Entailment Preserved (%)		83.18	42.61	42.77	4.92
N-gram Manhattan Distance (mean)	Uni-gram	3.34	2.65	4.76	9.86
	Bi-gram	2.39	1.59	2.26	2.33
	Tri-gram	2.34	1.60	2.13	1.94
Avg. number of Grammatical Errors		1.68	0.68	0.63	11.92

Table 14. Examples of original and paraphrased claims generated by several paraphrasing models.

Paraphrasing Model	Original Claim	Paraphrased Claim
$bart_{stanford}$	(1) Autologous transplantation of mesenchymal stem cells has worse graft function than induction therapy with anti-interleukin-2 receptor antibodies.	Autologous mesenchymal stem cell transplantation has a worse graft function than induction
	(2) A country's Vaccine Alliance (GAVI) eligibility is associated with accelerated adoption of the Hub vaccine.	Eligibility for the Global Alliance for Vaccines (GAVI) is linked to the
$bart_{eugene}$	(3) A breast cancer patient's capacity to metabolize tamoxifen has no effect on treatment outcome.	A breast cancer patient's ability to metabolize tamoxifen has no effect on treatment
	(4) Autologous transplantation of mesenchymal stem cells has worse graft function than induction therapy with anti-interleukin-2 receptor antibodies.	Autologous transplantation of mesenchymal stem cells has worse graft function than induction
alpaca	(5) Bronchial responsiveness is the same in the winter and summer seasons.	Bronchial responsiveness increases during the winter season but decreases in summer.
	(6) C. elegans germlines lose their immortal character when nuclear RNAi is activated.	C. elegans nucleotides are reduced to their less immortal state when nuclear transcription factor 1 (RT1) is activated in mitochondrial cells resulting in the death of eternal life as we know it today.

F ETHICAL CONSIDERATIONS

Scientific claim verification tools can help users navigate misinformation on a variety of complex and convoluted subjects. Our work proposes an approach which searches for vulnerabilities of such tools. While this approach may be used by malicious user to attack SCV tools, disclosure of the flaws of a system helps to ensure that newer designs are more robust. In fact, we subscribe to the generally-held notion that security through obscurity is not an effective practice, in accordance with Shannon's maxim, which states "one ought to design systems under the assumption that the enemy will immediately gain full familiarity with them." We are confident that this work will create awareness in the research community and stimulate the design of defense strategies and more robust automatic scientific claim verification tools. Thus, exposure of vulnerabilities is in this case ultimately to the benefit of the field of automatic scientific claim verification.