Improving Paraphrase Detection with the Adversarial Paraphrasing Task

Animesh Nighojkar, John Licato
Advancing Machine and Human Reasoning Lab
Department of Computer Science and Engineering
University of South Florida
Tampa, FL, USA
{anighojkar,licato}@usf.edu

Abstract

If two sentences have the same meaning, it should follow that they are equivalent in their inferential properties, i.e., each sentence should textually entail the other. However, many paraphrase datasets currently in widespread use rely on a sense of paraphrase based on word overlap and syntax. Can we teach them instead to identify paraphrases in a way that draws on the inferential properties of the sentences, and is not over-reliant on lexical and syntactic similarities of a sentence pair? We apply the adversarial paradigm to this question, and introduce a new adversarial method of dataset creation for paraphrase identification: the Adversarial Paraphrasing Task (APT), which asks participants to generate semantically equivalent (in the sense of mutually implicative) but lexically and syntactically disparate paraphrases. These sentence pairs can then be used both to test paraphrase identification models (which get barely random accuracy) and then improve their performance. To accelerate dataset generation, we explore automation of APT using T5, and show that the resulting dataset also improves accuracy. We discuss implications for paraphrase detection and release our dataset in the hope of making paraphrase detection models better able to detect sentence-level meaning equivalence.

1 Introduction

Although there are many definitions of 'paraphrase' in the NLP literature, most maintain that two sentences that are paraphrases have the same meaning or contain the same information. Pang et al. (2003) define paraphrasing as "expressing the same information in multiple ways" and Bannard and Callison-Burch (2005) call paraphrases "alternative ways of conveying the same information." Ganitkevitch et al. (2013) write that "paraphrases are differing textual realizations of the same meaning." A definition that seems to sufficiently encompass the others is given by Bhagat and Hovy (2013): "paraphrases are sentences or phrases that *use different wording to convey the same meaning*." However, even that definition is somewhat imprecise, as it lacks clarity on what it assumes 'meaning' means.

If paraphrasing is a property that can hold between sentence pairs,[1] then it is reasonable to assume that sentences that are paraphrases must have equivalent meanings at the sentence level (rather than exclusively at the levels of individual word meanings or syntactic structures). Here a useful test is that recommended by inferential role semantics or inferentialism (Boghossian, 1994; Peregrin, 2006), which suggests that the meaning of a statement s is grounded in its inferential properties: what one can infer from s and from what s can be inferred.

Building on this concept from inferentialism, we assert that if two sentences have the same inferential properties, then they should also be mutually implicative. Mutual Implication (MI) is a binary relationship between two sentences that holds when each sentence textually entails the other (i.e., bidirectional entailment). MI is an attractive way of operationalizing the notion of two sentences having "the same meaning," as it focuses on inferential relationships between sentences (properties of the sentences as wholes) instead of just syntactic or lexical similarities (properties of parts of the sentences). As such, we will assume in this paper that two sentences are paraphrases if and only if they are MI.[2] In NLP, modeling inferential relationships between sentences is the goal of the textual entailment, or natural language inference (NLI) tasks (Bowman et al., 2015). We test MI

[1] In this paper we study paraphrase between sentences, and do not address the larger scope of how our work might extend to paraphrasing between arbitrarily large text sequences.

[2] The notations used in this paper are listed in Table 1.

using the version of RoBERTa$_{large}$ released by Nie et al. (2020) trained on a combination of SNLI (Bowman et al., 2015), multiNLI (Williams et al., 2018), FEVER-NLI (Nie et al., 2019), and ANLI (Nie et al., 2020).

Owing to expeditious progress in NLP research, performance of models on benchmark datasets is 'plateauing' — with near-human performance often achieved within a year or two of their release — and newer versions, using a different approach, are constantly having to be created, for instance, GLUE (Wang et al., 2019) and SuperGLUE (Wang et al., 2020). The adversarial paradigm of dataset creation (Jia and Liang, 2017a,b; Bras et al., 2020; Nie et al., 2020) has been widely used to address this 'plateauing,' and the ideas presented in this paper draw inspiration from it. In the remainder of this paper, we apply the adversarial paradigm to the problem of paraphrase detection, and demonstrate the following **novel contributions**:

- We use the adversarial paradigm to create a new benchmark examining whether paraphrase detection models are assessing the meaning equivalence of sentences rather than being over-reliant on word-level measures. We do this by collecting paraphrases that are *MI* but are as lexically and syntactically disparate as possible (as measured by low BLEURT scores). We call this the Adversarial Paraphrasing Task (APT).
- We show that a SOTA language model trained on paraphrase datasets perform poorly on our benchmark. However, when further trained on our adversarially-generated datasets, their MCC scores improve by up to 0.307.
- We create an additional dataset by training a paraphrase generation model to perform our adversarial task, creating another large dataset that further improves the paraphrase detection models' performance.
- We propose a way to create a machine-generated adversarial dataset and discuss ways to ensure it does not suffer from the plateauing that other datasets suffer from.

2 Related Work

Paraphrase detection (given two sentences, predict whether they are paraphrases) (Zhang and Patrick, 2005; Fernando and Stevenson, 2008; Socher et al., 2011; Jia et al., 2020) is an important task in the field of NLP, finding downstream applications in machine translation (Callison-Burch et al., 2006; Apidianaki et al., 2018; Mayhew et al., 2020), text summarization, plagiarism detection (Hunt et al., 2019), question answering, and sentence simplification (Guo et al., 2018). Paraphrases have proven to be a crucial part of NLP and language education, with research showing that paraphrasing helps improve reading comprehension skills (Lee and Colln, 2003; Hagaman and Reid, 2008). Question paraphrasing is an important step in knowledge-based question answering systems for matching questions asked by users with knowledge-based assertions (Fader et al., 2014; Yin et al., 2015).

MI	Concept of mutual implication / bidirectional textual entailment)
MI	Property of being mutually implicative, as determined by our NLI model
APT	Adversarial Paraphrasing Task
APT	Property of passing the adversarial paraphrase test (see §3)
AP_H	Human-generated APT dataset
AP_{T5}	T5$_{base}$-generated APT dataset (Note that $AP_{T5} = AP^{M}_{T5} \cup AP^{Tw}_{T5}$)
AP^{M}_{T5}	MSRP subset of AP_{T5}
AP^{Tw}_{T5}	TwitterPPDB subset of AP_{T5}

Table 1: Notations used in the paper.

Paraphrase generation (given a sentence, generate its paraphrase) (Gupta et al., 2018) is an area of research benefiting paraphrase detection as well. Lately, many paraphrasing datasets have been introduced to be used for training and testing ML models for both paraphrase detection and generation. MSRP (Dolan and Brockett, 2005) contains 5801 sentence pairs, each labeled with a binary human judgment of paraphrase, created using heuristic extraction techniques along with an SVM-based classifier. These pairs were annotated by humans, who found 67% of them to be semantically equivalent. The English portion of PPDB (Ganitkevitch et al., 2013) contains over 220M paraphrase pairs generated by meaning-preserving syntactic transformations. Paraphrase pairs in PPDB 2.0 (Pavlick et al., 2015) include fine-grained entailment relations, word embedding similarities, and style annotations. TwitterPPDB (Lan et al., 2017) consists of 51,524 sentence pairs captured from Twitter by linking tweets through shared URLs. This ap-

proach's merit is its simplicity as it involves neither a classifier nor a human-in-the-loop to generate paraphrases. Humans annotate the pairs, giving them a similarity score ranging from 1 to 6.

ParaNMT (Wieting and Gimpel, 2018) was created by using neural machine translation to translate the English side of a Czech-English parallel corpus (CzEng 1.6 (Bojar et al., 2016)), generating more than 50M English-English paraphrases. However, ParaNMT's use of machine translation models that are a few years old harms its utility (Nighojkar and Licato, 2021), considering the rapid improvement in machine translation in the past few years. To rectify this, we use the google-translate library to translate the Czech side of roughly 300k CzEng2.0 (Kocmi et al., 2020) sentence pairs ourselves. We call this dataset *ParaParaNMT* (PPNMT for short, where the extra *para-* prefix reflects its similarity to, and conceptual derivation from, ParaNMT).

Some work has been done in improving the quality of paraphrase detectors by training them on a dataset with more lexical and syntactic diversity. Thompson and Post (2020) propose a paraphrase generation algorithm that penalizes the production of n-grams present in the source sentence. Our approach to doing this is with the APT, but this is something worth exploring. Sokolov and Filimonov (2020) use a machine translation model to generate paraphrases much like ParaNMT. An interesting application of paraphrasing has been discussed by Mayhew et al. (2020) who, given a sentence in one language, generate a diverse set of correct translations (paraphrases) that humans are likely to produce. In comparison, our work is focused on generating adversarial paraphrases that are likely to deceive a paraphrase detector, and models trained on the adversarial datasets we produce can be applied to Mayhew et al.'s work too.

ANLI (Nie et al., 2020), a dataset designed for Natural Language Inference (NLI) (Bowman et al., 2015), was collected via an adversarial human-and-model-in-the-loop procedure where humans are given the task of duping the model into making a wrong prediction. The model then tries to learn how not to make the same mistakes. AFLite (Bras et al., 2020) adversarially filters dataset biases making sure that the models are not learning those biases. They show that model performance on SNLI (Bowman et al., 2015) drops from 92% to 62% when biases were filtered out. However, their approach is to filter the dataset, which reduces its size, making model training more difficult. Our present work tries instead to generate adversarial examples to *increase* dataset size. Other examples of adversarial datasets in NLP include work done by Jia and Liang (2017a); Zellers et al. (2018, 2019). Perhaps the closest to our work is PAWS (Zhang et al., 2019), short for Paraphrase Adversaries from Word Scrambling. The idea behind PAWS is to create a dataset that has a high lexical overlap between sentence pairs without them being 'paraphrases.' It has 108k paraphrase and non-paraphrase pairs with high lexical overlap pairs generated by controlled word swapping and back-translation, and human raters have judged whether or not they are paraphrases. Including PAWS in the training data has shown the state-of-the-art models' performance to jump from 40% to 85% on PAWS's test split. In comparison to the present work, PAWS does not explicitly incorporate inferential properties, and we seek paraphrases *minimizing* lexical overlap.

3 Adversarial Paraphrasing Task (APT)

Semantic Textual Similarity (STS) measures the degree of semantic similarity between two sentences. Popular approaches to calculating STS include BLEU (Papineni et al., 2002), BertScore (Zhang et al., 2020), and BLEURT (Sellam et al., 2020). BLEURT is a text generation metric building on BERT's (Devlin et al., 2019) contextual word representations. BLEURT is *warmed-up* using synthetic sentence pairs and then fine-tuned on human ratings to generalize better than BERTScore (Zhang et al., 2020). Given any two sentences, BLEURT assigns them a similarity score (usually between -2.2 to 1.1). However, high STS scores do not necessarily predict whether two sentences have equivalent meanings. Consider the sentence pairs in Table 3, highlighting cases where STS and paraphrase appear to misalign. The existence of such cases suggests a way to advance automated paraphrase detection: *through an adversarial benchmark consisting of sentence pairs that have the same MI-based meaning, but have BLEURT scores that are as low as possible.* This is the motivation behind what we call the Adversarial Paraphrasing Task (APT), which has two components:

1. *Similarity of meaning*: Checked through MI (Section 1). We assume if two sentences are MI (Mutually Implicative), they are semantically equivalent and thus paraphrases. Note

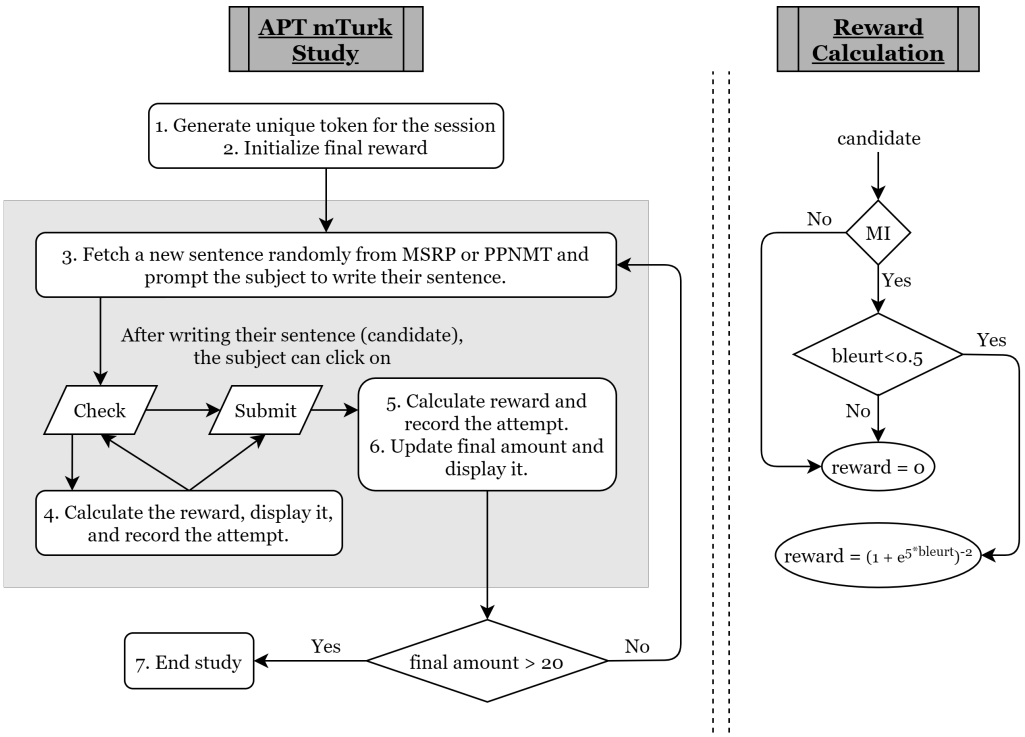


Figure 1: The mTurk study and the reward calculation. We automatically end the study when a subject earns a total of $20 to ensure variation amongst subjects.

that MI is a binary relationship, so this APT component does not bring any quantitative variation but is more like a qualifier test for APT. All APT sentence pairs are MI.

2. *Dissimilarity of structure*: Measured through BLEURT, which assigns each sentence pair a score quantifying how lexically and syntactically similar the two sentences are.

3.1 Manually Solving APT

To test the effectiveness of APT in guiding the generation of mutually implicative but lexically and syntactically disparate paraphrases for a given sentence, we designed an Amazon Mechanical Turk (mTurk) study (Figure 1). Given a starting sentence, we instructed participants to "[w]rite a sentence that is the same in meaning as the given sentence but as structurally different as possible. Your sentence should be such that you can infer the given sentence from it AND vice-versa. It should be sufficiently different from the given sentence to get any reward for the submission. For example, a simple synonym substitution will most likely not work." The sentences given to the participants came from MSRP and PPNMT (Section 1). Both of these datasets have pairs of sentences in each row, and we took only the first one to present to the participants. Neither of these datasets has duplicate sentences by design. Every time a sentence was selected, a random choice was made between MSRP and PPNMT, thus ensuring an even distribution of sentences from both datasets.

Each attempt was evaluated separately using Equation 1, where mi is 1 when the sentences are MI and 0 otherwise:

$$reward = \frac{mi}{(1 + e^{5*bleurt})^2} \quad (1)$$

This formula was designed to ensure (1) the maximum reward per submission was $1, and (2) no reward was granted for sentence pairs that are non-MI or have BLEURT > 0.5. Participants were encouraged to frequently revise their sentences and click on a 'Check' button which showed them the reward amount they would earn if they submitted this sentence. Once the 'Check' button was clicked, the participant's reward was evaluated (see Figure 1) and the sentence pair added to AP_H (regardless of whether it was APT). If 'Submit' was clicked, their attempt was rewarded based on Equation 1.

The resulting dataset of sentence pairs, which we call AP_H (Adversarial Paraphrase by Humans), consists of 5007 human-generated sentence pairs, both MI and non-MI (see Table 2). Humans were able to generate APT paraphrases for 75.48% of

Dataset	Total attempts	APT attempts	MI attempts	non-MI attempts	Unique sentences	APT uniques	MI uniques	non-MI uniques
AP_H	5007	2659 **53.10%**	3232 64.55%	1775 35.45%	1631	1231 **75.48%**	1338 82.04%	293 17.96%
AP^M_{T5}	62,986	3836 **6.09%**	37,511 59.55%	25,475 40.44%	4072	2288 **56.19%**	4045 99.34%	3115 76.50%
AP^{Tw}_{T5}	75,011	6454 **8.60%**	17,074 22.76%	57,937 77.24%	4328	3670 **84.80%**	4131 95.45%	4230 97.74%

Table 2: Proportion of sentences generated by humans (AP_H) and T5$_{base}$ (AP_{T5}). "Attempts" shows the number of attempts the participant made and "Uniques" shows the number of source sentences from the dataset that the performer's attempts fall in that category on. For instance, 1631 unique sentences were presented to humans, who made a total of 5007 attempts to pass APT and were able to do so for 2659 attempts which amounted to 1231 unique source sentences that could be paraphrased to pass APT.

the sentences presented to them and only 53.1% of attempts were APT, showing that the task is difficult even for humans. Note that 'MI attempts' and 'MI uniques' are supersets of 'APT attempts' and 'APT uniques,' respectively.

3.2 Automatically Solving APT

Since human studies can be time-consuming and costly, we trained a paraphrase generator to perform APT. We used T5$_{base}$ (Raffel et al., 2020), as it achieves SOTA on paraphrase generation (Niu et al., 2020; Bird et al., 2020; Li et al., 2020) and trained it on TwitterPPDB (Section 2). Our hypothesis was that if T5$_{base}$ is trained to maximize the APT reward (Equation 1), its generated sentences will be more likely to be APT. We generated paraphrases for sentences in MSRP and those in TwitterPPDB itself, hoping that since T5$_{base}$ is trained on TwitterPPDB, it would generate better paraphrases (MI with lower BLEURT) for sentences coming from there. The proportion of sentences generated by T5$_{base}$ is shown in Table 2. We call this dataset AP_{T5}, the generation of which involved two phases:

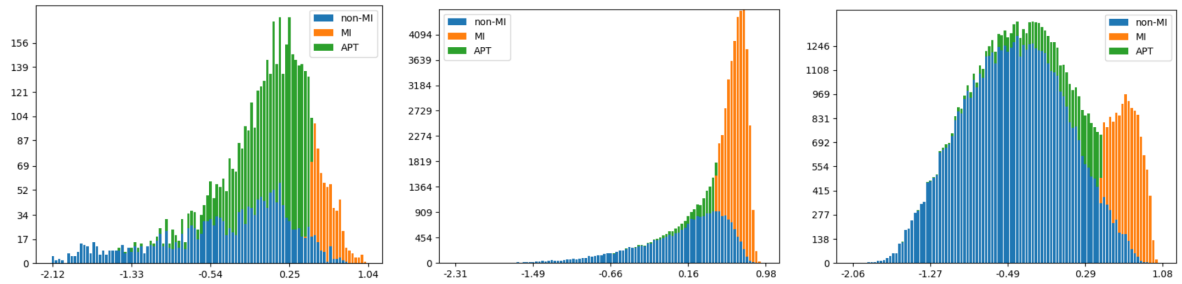
Training: To adapt T5$_{base}$ for APT, we implemented a custom loss function obtained from dividing the cross-entropy loss per batch by the total reward (again from Equation 1) earned from the model's paraphrase generations for that batch, provided the model was able to reach a reward of at least 1. If not, the loss was equal to just the cross-entropy loss. We trained T5$_{base}$ on TwitterPPDB for three epochs; each epoch took about 30 hours on one NVIDIA Tesla V100 GPU due to the CPU bound BLEURT component. More epochs *may* help get better results, but our experiments showed that loss plateaus after three epochs.

Generation: Sampling, or randomly picking a next word according to its conditional probability distribution, introduces non-determinism in language generation. Fan et al. (2018) introduce top-k sampling, which filters k most likely next words, and the probability mass is redistributed among only those k words. Nucleus sampling (or top-p sampling) (Holtzman et al., 2020) reduces the options to the smallest possible set of words whose cumulative probability exceeds p, and the probability mass is redistributed among this set of words. Thus, the set of words changes dynamically according to the next word's probability distribution. We use a combination of top-k and top-p sampling with $k = 120$ and $p = 0.95$ in the interest of lexical and syntactic diversity in the paraphrases. For each sentence in the source dataset (MSRP[3] and TwitterPPDB for AP^M_{T5} and AP^{Tw}_{T5} respectively), we perform five iterations, in each of which, we generate ten sentences. If at least one of these ten sentences passes APT, we continue to the next source sentence after recording all attempts and classifying them as MI or non-MI. If no sentence in a maximum of 50 attempts passes APT, we record all attempts nonetheless, and move on to the next source sentence. For each increasing iteration for a particular source sentence, we increase k by 20, but we also reduce p by 0.05 to avoid vague guesses. Note the distribution of MI and non-MI in the source datasets does not matter because we use only the first sentence from the sentence pair.

3.3 Dataset Properties

T5$_{base}$ trained with our custom loss function generated APT-passing paraphrases for (56.19%) of starting sentences. This is higher than we initially expected, considering how difficult APT proved to be for humans (Table 2). Noteworthy is that

[3]We use the official train split released by Dolan and Brockett (2005) containing 4076 sentence pairs.



(a) AP_H (b) AP^M_{T5} (c) AP^{Tw}_{T5}

Figure 2: BLEURT distributions on adversarial datasets. All figures divide the range of observed scores into 100 bins. Note that APT sentence pairs are also MI, whereas those labeled 'MI' are not APT.

only 6.09% of T5$_{base}$'s attempts were APT. This does not mean that the remaining 94% of attempts can be discarded, since they amounted to the negative examples in the dataset. Since we trained it on TwitterPPDB itself, we expected that T5$_{base}$ would generate better paraphrases, as measured by a higher chance of passing APT on TwitterPPDB, than any other dataset we tested. This is supported by the data in Table 2, which shows that T5$_{base}$ was able to generate an APT passing paraphrase for 84.8% of the sentences in TwitterPPDB.

The composition of the three adversarial datasets can be found in Table 2. These metrics are useful to understand the capabilities of T5$_{base}$ as a paraphrase generator and the "paraphrasability" of sentences in MSRP and TwitterPPDB. For instance, T5$_{base}$'s attempts on TwitterPPDB tend to be MI much less frequently than those on MSRP and human's attempts on MSRP + PPNMT. This might be because in an attempt to generate syntactically dissimilar sentences, the T5$_{base}$ paraphraser also ended up generating many semantically dissimilar ones as well.

To visualize the syntactic and lexical disparity of paraphrases in the three adversarial datasets, we present their BLEURT distributions in Figure 2. As might be expected, the likelihood of a sentence pair being MI increases as BLEURT score increases (recall that APT-passing sentence pairs are simply MI pairs with BLEURT scores $<= 0.5$), but Figure 2 shows that the shape of this increase is not straightforward, and differs among the three datasets.

As might be expected, humans are much more skilled at APT than T5$_{base}$, as shown by the fact that the paraphrases they generated have much lower mean BLEURT scores (Figure 2), and the ratio of APT vs non-APT sentences is much higher (Table 2). As we saw earlier, when T5$_{base}$ wrote paraphrases that were low on BLEURT, they tended to become non-MI (e.g., line 12 in Table 3). However, T5$_{base}$ did generate more APT-passing sentences with a lower BLEURT on Twitter-PPDB than on MSRP, which may be a result of overfitting T5$_{base}$ on TwitterPPDB. Furthermore, all three adversarial datasets have a distribution of MI and non-MI sentence pairs balanced enough to train a model to identify paraphrases.

Table 3 has examples from AP_H and AP_{T5} showing the merits and shortcomings of T5, BLEURT, and RoBERTa$_{large}$ (the MI detector used). Some observations from Table 3 include:

- *Lines 1 and 3:* BLEURT did not recognize the paraphrases, possibly due to the differences in words used. RoBERTa$_{large}$ however, gave the correct MI prediction (though it is worth noting that the sentences in line 1 are questions, rather than truth-apt propositions).

- *Line 4:* RoBERTa$_{large}$ and BLEURT (to a large extent since it gave it a score of 0.4) did not recognize that the idiomatic phrase 'break a leg' means 'good luck' and not 'fracture.'

- *Lines 6 and 12:* There is a loss of information going from the first sentence to the second and BLEURT and MI both seem to have understood the difference between summarization and paraphrasing.

- *Line 7:* T5 not only understood the scores but also managed to paraphrase it in such a way that was not syntactically and lexically similar, just as we wanted T5 to do when we fine-tuned it.

- *Line 9:* T5$_{base}$ knows that Fort Lauderdale is in Florida but RoBERTa$_{large}$ does not.

No.	Source Dataset	Source Sentence	Attempt	BLEURT	MI
		APH			
1	PPNMT	So, can we please get out of here?	So is it okay if we please go?	-0.064	1
2	PPNMT	You're crying.	I did not cry	-1.366	0
3	PPNMT	Treatment successful.	The treatment was succesful.	-0.871	1
4	PPNMT	Break a leg!	Fracture a leg!	0.408	1
5	MSRP	Two years later, the insurance coverage would begin.	The insurance will start in two years	0.281	1
6	MSRP	Evacuation went smoothly, although passengers weren't told what was going on, Hunt said.	Hunt told that Evacuation went smoothly.	-0.298	0
		$APT5$			
7	MSRP	Friday, Stanford (47-15) blanked the Gamecocks 8-0.	Stanford (47-15) won 8-0 over the Gamecocks on Friday.	0.206	1
8	MSRP	Revenue in the first quarter of the year dropped 15 percent from the same period a year earlier.	Revenue declined 15 percent in the first quarter of the year from the same period a year earlier.	0.698	1
9	MSRP	A federal magistrate in Fort Lauderdale ordered him held without bail.	In Fort Lauderdale, Florida, a federal magistrate ordered him held without bail.	0.635	0
10	TP	16 innovations making a difference for poor communities around the world.	16 innovative ideas that tackle poverty around the world.	0.317	1
11	TP	This is so past the bounds of normal or acceptable .	This is so beyond the normal or acceptable boundaries.	0.620	1
12	TP	The creator of Atari has launched a new VR company called Modal VR.	Atari creator is setting up a new VR company!	0.106	0

Table 3: Examples from adversarial datasets. The source dataset (TP short for TwitterPPDB) tells which dataset the sentence pair comes from (and whether it is in $APM5$ or $APTw5$ for $APT5$). All datasets have APT passing and failing MI and non-MI sentence pairs.

Dataset	Total	MI		non-MI	
APH-train	3746	2433	64.95%	1313	35.05%
APH-test	1261	799	63.36%	462	36.64%
MSRP-train	4076	2753	67.54%	1323	32.46%
MSRP-test	1725	1147	66.50%	578	33.50%

Table 4: Distribution of MI and non-MI pairs.

Test Set	RoBERTa$_{base}$		Random	
	MCC	F1	MCC	F1
MSRP-train	0.349	0.833	0	0.806
MSRP-test	0.358	0.829	0	0.799
APH	0.222	0.746	0	0.784
APH-test	0.218	0.743	0	0.777

Table 5: Performance of RoBERTa$_{base}$ trained on just TwitterPPDB (no adversarial datasets) vs. random prediction.

Training Dataset TwitterPPDB +	Size	APH		APH-test	
		MCC	F1	MCC	F1
APH-train	46k			**0.440**	0.809
$APM5$	106k	0.410	0.725	0.369	0.705
APH-train + $APM5$	109k			**0.516**	0.828
$APTw5$	117k	0.433	0.772	0.422	0.765
APH-train + $APTw5$	121k			0.488	0.812
$APT5$	180k	0.461	0.731	0.437	0.716
APH-train + $APT5$	184k			**0.525**	0.816

Table 6: Performance of RoBERTa$_{base}$ trained on adversarial datasets. Size is the number of training examples in the dataset rounded to nearest 1000.

4 Experiments

To quantify our datasets' contributions, we designed experiment setups wherein we trained RoBERTa$_{base}$ (Liu et al., 2019) for paraphrase detection on a combination of TwitterPPDB and our datasets as training data. RoBERTa was chosen for its generality, as it is a commonly used model in current NLP work and benchmarking, and currently achieves SOTA or near-SOTA results on a majority of NLP benchmark tasks (Wang et al., 2019, 2020; Chen et al., 2021).

For each source sentence, multiple paraphrases may have been generated. Hence, to avoid data leakage, we created a train-test split on APH such that all paraphrases generated using a given source sentence will be either in APH-train or in APH-test, but never in both. Note that APH is not balanced as seen in Table 2. Table 4 shows the distribution of MI and non-MI pairs in APH-train and APH-test and 'MI attempts' and 'non-MI attempts' columns of Table 2 show the same for other adversarial datasets. The test sets used were APH wherever APH-train was not a part of the training data and APH-test in every case.

Does RoBERTa$_{base}$ do well on APT_H? RoBERTa$_{base}$ was trained on each training dataset (90% training data, 10% validation data) for five epochs with a batch size of 32 with the training and validation data shuffled, and the trained model was tested on APT_H and APT_H-test. The results of this are shown in Table 6. Note that since the number of MI and non-MI sentences in all the datasets is imbalanced, Matthew's Correlation Coefficient (MCC) is a more appropriate performance measure than accuracy (Boughorbel et al., 2017).

Our motivation behind creating an adversarial dataset was to improve the performance of paraphrase detectors by ensuring they recognize paraphrases with low lexical overlap. To demonstrate the extent of their inability to do so, we first compare the performance of RoBERTa$_{base}$ trained only on TwitterPPDB on specific datasets as shown Table 5. Although the model performs slightly well on MSRP, it does barely better than a random prediction on APT_H, thus showing that identifying adversarial paraphrases created using APT is non-trivial for paraphrase identifiers.

Do human-generated adversarial paraphrases improve paraphrase detection? We introduce APT_H-train to the training dataset along with TwitterPPDB. This improves the MCC by 0.222 even though APT_H-train constituted just 8.15% of the entire training dataset, the rest of which was TwitterPPDB (Table 6). This shows the effectiveness of human-generated paraphrases, as is especially impressive given the size of APT_H-train compared to TwitterPPDB.

Do machine-generated adversarial paraphrases improve paraphrase detection? We set out to test the improvement brought by APT_{T5}, of which we have two versions. Adding APT^{M}_{T5} to the training set was not as effective as adding APT_H-train, increasing MCC by 0.188 on APT_H and 0.151 on APT_H-test, thus showing us that T5$_{base}$, although was able to clear APT, lacked the quality which human paraphrases possessed. This might be explained by Figure 2 — since APT^{M}_{T5} does not have many sentences with low BLEURT, we cannot expect a vast improvement in RoBERTa$_{base}$'s performance on sentences with BLEURT as low as in APT_H.

Since we were not necessarily testing T5$_{base}$'s performance — and we had trained T5$_{base}$ on TwitterPPDB — we used the trained model to perform APT on TwitterPPDB itself. Adhering to expectations, training RoBERTa$_{base}$ (the paraphrase detector) with APT^{Tw}_{T5} yielded higher MCCs. Note that none of the sentences are common between APT^{Tw}_{T5} and APT_H since APT_H is built on MSRP and PPNMT and the fact that the model got this performance when trained on APT^{Tw}_{T5} is a testimony to the quality and contribution of APT.

Combining these results, we can conclude that although machine-generated datasets like APT_{T5} can help paraphrase detectors improve themselves, a smaller dataset of human-generated adversarial paraphrases improved performance more. Overall, however, the highest MCC (0.525 in Table 6) is obtained when TwitterPPDB is combined with all three adversarial datasets, suggesting that the two approaches nicely complement each other.

5 Discussions and Conclusions

This paper introduced APT (Adversarial Paraphrasing Task), a task that uses the adversarial paradigm to generate paraphrases consisting of sentences with equivalent (sentence-level) meanings, but differing lexical (word-level) and syntactical similarity. We used APT to create a human-generated dataset / benchmark (APT_H) and two machine-generated datasets (APT^{M}_{T5} and APT^{Tw}_{T5}). Our goal was to effectively augment how paraphrase detectors are trained, in order to make them less reliant on word-level similarity. In this respect, the present work succeeded: we showed that RoBERTa$_{base}$ trained on TwitterPPDB performed poorly on APT benchmarks, but this performance was increased significantly when further trained on either our human- or machine-generated datasets. The code used in this paper along with the dataset has been released in a publicly-available repository.[4]

Paraphrase detection and generation have broad applicability, but most of their potential lies in areas in which they still have not been substantially applied. These areas range from healthcare (improving accessibility to medical communications or concepts by automatically generating simpler language), writing (changing the writing style of an article to match phrasing a reader is better able to understand), and education (simplifying the language of a scientific paper or educational lesson to

[4] https://github.com/Advancing-Machine-Human-Reasoning-Lab/apt

make it easier for students to understand). Thus, future research into improving their performance can be very valuable. But approaches to paraphrase that treat it as no more than a matter of detecting word similarity overlap will not suffice for these applications. Rather, the meanings of sentences are properties of the sentences as a whole, and are inseparably tied to their inferential properties. Thus, our approaches to paraphrase detection and generation must follow suit.

The adversarial paradigm can be used to dive deeper into comparing how humans and SOTA language models understand sentence meaning, as we did with APT. Furthermore, automatic generation of adversarial datasets has much unrealized potential; e.g., different datasets, paraphrase generators, and training approaches can be used to generate future versions of AP_{T5} in order to produce APT passing sentence pairs with lower lexical and syntactic similarities (as measured not only by BLEURT, but also by future state-of-the-art STS metrics). The idea of more efficient automated adversarial task performance is particularly exciting, as it points to a way language models can improve themselves while avoiding prohibitively expensive human participant fees.

Finally, the most significant contribution of this paper, APT, presents a dataset creation method for paraphrases that will not saturate because as the models get better at identifying paraphrases, we will improve paraphrase generation. As models get better at generating paraphrases, we can make APT harder (e.g., by reducing the BLEURT threshold of < 0.5). One might think of this as students in a class who come up with new ways of copying their assignments from sources as plagiarism detectors improved. That brings us to one of the many applications of paraphrases: plagiarism generation and detection, which inherently is an adversarial activity. Until plagiarism detectors are trained on adversarial datasets themselves, we cannot expect them to capture human levels of adversarial paraphrasing.

Acknowledgements

This material is based upon work supported by the Air Force Office of Scientific Research under award numbers FA9550-17-1-0191 and FA9550-18-1-0052. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the United States Air Force. We would also like to thank Antonio Laverghetta Jr. and Jamshidbek Mirzakhalov for their helpful suggestions while writing this paper, and Gokul Shanth Raveendran and Manvi Nagdev for helping with the website used for the mTurk study.

References

Marianna Apidianaki, Guillaume Wisniewski, Anne Cocos, and Chris Callison-Burch. 2018. Automated paraphrase lattice creation for hyter machine translation evaluation. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 480–485.

Colin Bannard and Chris Callison-Burch. 2005. Paraphrasing with bilingual parallel corpora. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL'05)*, pages 597–604.

Rahul Bhagat and Eduard Hovy. 2013. Squibs: What is a paraphrase? *Computational Linguistics*, 39(3):463–472.

Jordan J. Bird, Anikó Ekárt, and Diego R. Faria. 2020. Chatbot interaction with artificial intelligence: Human data augmentation with t5 and language transformer ensemble for text classification.

Paul A. Boghossian. 1994. Inferential role semantics and the analytic/synthetic distinction. *Philosophical Studies: An International Journal for Philosophy in the Analytic Tradition*, 73(2/3):109–122.

Ondřej Bojar, Ondřej Dušek, Tom Kocmi, Jindřich Libovickỳ, Michal Novák, Martin Popel, Roman Sudarikov, and Dušan Variš. 2016. Czeng 1.6: enlarged czech-english parallel corpus with processing tools dockered. In *International Conference on Text, Speech, and Dialogue*, pages 231–238. Springer.

Sabri Boughorbel, Fethi Jarray, and Mohammed El-Anbari. 2017. Optimal classifier for imbalanced data using matthews correlation coefficient metric. *PloS one*, 12(6):e0177678–e0177678. 28574989[pmid].

Samuel R. Bowman, Gabor Angeli, Christopher Potts, and Christopher D. Manning. 2015. A large annotated corpus for learning natural language inference.

Ronan Le Bras, Swabha Swayamdipta, Chandra Bhagavatula, Rowan Zellers, Matthew E. Peters, Ashish Sabharwal, and Yejin Choi. 2020. Adversarial filters of dataset biases.

Chris Callison-Burch, Philipp Koehn, and Miles Osborne. 2006. Improved statistical machine translation using paraphrases. In *Proceedings of the*

Human Language Technology Conference of the NAACL, Main Conference, pages 17–24, New York City, USA. Association for Computational Linguistics.

Ben Chen, Bin Chen, Dehong Gao, Qijin Chen, Chengfu Huo, Xiaonan Meng, Weijun Ren, and Yang Zhou. 2021. Transformer-based language model fine-tuning methods for covid-19 fake news detection.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding.

Bill Dolan and Chris Brockett. 2005. Automatically constructing a corpus of sentential paraphrases. In *Third International Workshop on Paraphrasing (IWP2005)*. Asia Federation of Natural Language Processing.

Anthony Fader, Luke Zettlemoyer, and Oren Etzioni. 2014. Open question answering over curated and extracted knowledge bases. In *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '14, page 1156–1165, New York, NY, USA. Association for Computing Machinery.

Angela Fan, Mike Lewis, and Yann Dauphin. 2018. Hierarchical neural story generation.

Samuel Fernando and Mark Stevenson. 2008. A semantic similarity approach to paraphrase detection. In *Proceedings of the 11th annual research colloquium of the UK special interest group for computational linguistics*, pages 45–52.

Juri Ganitkevitch, Benjamin Van Durme, and Chris Callison-Burch. 2013. Ppdb: The paraphrase database. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 758–764.

Han Guo, Ramakanth Pasunuru, and Mohit Bansal. 2018. Dynamic multi-level multi-task learning for sentence simplification.

Ankush Gupta, Arvind Agarwal, Prawaan Singh, and Piyush Rai. 2018. A deep generative framework for paraphrase generation. *Proceedings of the AAAI Conference on Artificial Intelligence*, 32(1).

Jessica L. Hagaman and Robert Reid. 2008. The effects of the paraphrasing strategy on the reading comprehension of middle school students at risk for failure in reading. *Remedial and Special Education*, 29(4):222–234.

Ari Holtzman, Jan Buys, Li Du, Maxwell Forbes, and Yejin Choi. 2020. The curious case of neural text degeneration.

E. Hunt, R. Janamsetty, C. Kinares, C. Koh, A. Sanchez, F. Zhan, M. Ozdemir, S. Waseem, O. Yolcu, B. Dahal, J. Zhan, L. Gewali, and P. Oh. 2019. Machine learning models for paraphrase identification and its applications on plagiarism detection. In *2019 IEEE International Conference on Big Knowledge (ICBK)*, pages 97–104.

Robin Jia and Percy Liang. 2017a. Adversarial examples for evaluating reading comprehension systems.

Robin Jia and Percy Liang. 2017b. Adversarial examples for evaluating reading comprehension systems. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2021–2031, Copenhagen, Denmark. Association for Computational Linguistics.

Xin Jia, Wenjie Zhou, Xu Sun, and Yunfang Wu. 2020. How to ask good questions? try to leverage paraphrases. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6130–6140, Online. Association for Computational Linguistics.

Tom Kocmi, Martin Popel, and Ondrej Bojar. 2020. Announcing czeng 2.0 parallel corpus with over 2 gigawords. *arXiv preprint arXiv:2007.03006*.

Wuwei Lan, Siyu Qiu, Hua He, and Wei Xu. 2017. A continuously growing dataset of sentential paraphrases. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1224–1234, Copenhagen, Denmark. Association for Computational Linguistics.

Steven Lee and Theresa Colln. 2003. The effect of instruction in the paraphrasing strategy on reading fluency and comprehension.

Eric Li, Jingyi Su, Hao Sheng, and Lawrence Wai. 2020. Agent zero: Zero-shot automatic multiple-choice question generation for skill assessments.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach.

Stephen Mayhew, Klinton Bicknell, Chris Brust, Bill McDowell, Will Monroe, and Burr Settles. 2020. Simultaneous translation and paraphrase for language education. In *Proceedings of the ACL Workshop on Neural Generation and Translation (WNGT). ACL.*

Yixin Nie, Haonan Chen, and Mohit Bansal. 2019. Combining fact extraction and verification with neural semantic matching networks. In *Association for the Advancement of Artificial Intelligence (AAAI).*

Yixin Nie, Adina Williams, Emily Dinan, Mohit Bansal, Jason Weston, and Douwe Kiela. 2020. Adversarial nli: A new benchmark for natural language understanding.

Animesh Nighojkar and John Licato. 2021. Mutual implication as a measure of textual equivalence. *The International FLAIRS Conference Proceedings*, 34.

Tong Niu, Semih Yavuz, Yingbo Zhou, Huan Wang, Nitish Shirish Keskar, and Caiming Xiong. 2020. Unsupervised paraphrase generation via dynamic blocking.

Bo Pang, Kevin Knight, and Daniel Marcu. 2003. Syntax-based alignment of multiple translations: Extracting paraphrases and generating new sentences. In *Proceedings of the 2003 Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics*, pages 181–188.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.

Ellie Pavlick, Pushpendre Rastogi, Juri Ganitkevitch, Benjamin Van Durme, and Chris Callison-Burch. 2015. Ppdb 2.0: Better paraphrase ranking, fine-grained entailment relations, word embeddings, and style classification. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 425–430.

Jaroslav Peregrin. 2006. Meaning as an inferential role. *Erkenntnis*, 64(1):1–35.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer.

Thibault Sellam, Dipanjan Das, and Ankur P. Parikh. 2020. Bleurt: Learning robust metrics for text generation.

Richard Socher, Eric H Huang, Jeffrey Pennin, Christopher D Manning, and Andrew Y Ng. 2011. Dynamic pooling and unfolding recursive autoencoders for paraphrase detection. In *Advances in neural information processing systems*, pages 801–809.

Alex Sokolov and Denis Filimonov. 2020. Neural machine translation for paraphrase generation.

Brian Thompson and Matt Post. 2020. Paraphrase generation as zero-shot multilingual translation: Disentangling semantic similarity from lexical and syntactic diversity.

Alex Wang, Yada Pruksachatkun, Nikita Nangia, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R. Bowman. 2020. Superglue: A stickier benchmark for general-purpose language understanding systems.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R. Bowman. 2019. Glue: A multi-task benchmark and analysis platform for natural language understanding.

John Wieting and Kevin Gimpel. 2018. ParaNMT-50M: Pushing the limits of paraphrastic sentence embeddings with millions of machine translations. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 451–462, Melbourne, Australia. Association for Computational Linguistics.

Adina Williams, Nikita Nangia, and Samuel R. Bowman. 2018. A broad-coverage challenge corpus for sentence understanding through inference.

Pengcheng Yin, Nan Duan, Ben Kao, Junwei Bao, and Ming Zhou. 2015. Answering questions with complex semantic constraints on open knowledge bases. In *Proceedings of the 24th ACM International on Conference on Information and Knowledge Management*, CIKM '15, page 1301–1310, New York, NY, USA. Association for Computing Machinery.

Rowan Zellers, Yonatan Bisk, Roy Schwartz, and Yejin Choi. 2018. Swag: A large-scale adversarial dataset for grounded commonsense inference.

Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. 2019. Hellaswag: Can a machine really finish your sentence?

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. 2020. Bertscore: Evaluating text generation with bert.

Yitao Zhang and Jon Patrick. 2005. Paraphrase identification by text canonicalization. In *Proceedings of the Australasian Language Technology Workshop 2005*, pages 160–166.

Yuan Zhang, Jason Baldridge, and Luheng He. 2019. Paws: Paraphrase adversaries from word scrambling.