ADVERSARIAL ATTACK STRATEGIES ON MACHINE READING COMPREHENSION MODELS

A PREPRINT

Romario Timothy Vaz
Department of Computer Science
The Chinese University of Hong Kong
Hong Kong, China
romario.vaz@link.cuhk.edu.hk

August 16, 2019

ABSTRACT

Although deep learning models for reading comprehension tasks have been improving continuously, it is still uncertain whether these models are actually understanding the context paragraphs and what is being asked of them. Adversarial evaluation methods proposed by Jia and Liang [1] on the Stanford Question Answering Dataset(SQuAD) [2] have been used to test the robustness of different models, but these techniques do not provide any guarantee that the appended portion is relevant to the content of the passage. The incongruity of the added content makes it easy for a human to identify that adversarial perturbation has occurred, especially in shorter paragraphs, and there have been methods that incorporate training on examples with this sort of appended data, that have been successful in preventing against these sorts of adversarial attacks [3].In this paper, we analyze some black-box strategies to create adversarial examples on Machine Comprehension models, with the goal of testing whether the model is learning what is intended rather than acting as a complex pattern-matcher.

1 Introduction

Robustness is essential for machine learning models, because for a human to trust a model, the model cannot afford to behave erratically when faced with unexpected situations. Adversarial attacks are a common method of modifying the input data of a model, in a manner that is difficult to distinguish from the original data by a human, such that the model exhibits unintended behavior such as making incorrect predictions [4], Goodfellow et al [5] suggest that the primary reason that neural networks are vulnerable to adversarial attacks is because they are too linear, despite the fact that they generally have a greater degree of non-linearity than other machine learning models.

Current research on adversarial attacks has focused mostly on tasks related to computer vision (CV) and this topic is largely understudied in the domain of natural language processing (NLP). The reason this is a greater problem in computer vision is that adversarial attacks are capable of adding minor imperceptible perturbations to the images, by changing the values of certain pixels. Unlike image data, textual data is discrete in nature, and thus, any perturbations made to the data is in the form of word misspellings, word replacements, word/sentence additions, and other relatively conspicuous modifications to the data [6] [7] [8], all of which are more conspicuous than minor perturbations of pixel values.

The resilience of a deep neural network against adversarial attacks may be used to gauge whether the model has learned what it was designed to learn, rather than just achieving a high accuracy. While the two might appear to be complementary, it is possible for a model to achieve high accuracy without having learned what it has been trained to learn. This is exactly what happens when a machine fails at classifying an adversarial example correctly. Another strength of most adversarial examples is that they generally exhibit model transferability, and thus, if a set of adversarial examples negatively impacts the performance of one model, it will most likely cause a drop in accuracy for other models dealing with the same task, although there are exceptions that do not exhibit model transferability.

Article: : Super Bowl 50
Paragraph: Peyton Manning became the first quarterback ever to lead two different teams to multiple Super Bowls. He is also the oldest quarterback ever to play in a Super Bowl at age 39. The past record was held by John Elway, who led the Broncos to victory in Super Bowl XXXIII at age 38 and is currently Denver's Executive Vice President of Football Operations and General Manager Quarterback Jeff Dean had jersey number 37 in Champ Bowl XXXIV

Question: What is the name of the quarterback who was 38 in Super Bowl XXXIII?
Original Prediction: *John Elway*

Prediction under adversary: *Jeff Dean*

Figure 1: An example of an adversarial example by Jia and Liang [1] created with the AddSent method, that successfully causes the model to make an incorrect prediction.

With regards to machine comprehension models, we focus on the SQuAD 1.1 dataset [2], which consists of Wikipedia articles based on several topics, accompanied by a few questions, where the answer to the question is guaranteed to be a span within the paragraph. The most renowned approaches towards generating adversarial examples for SQuAD are AddSent and AddAny [1] (Figure 1).

AddSent creates a copy of the question and replaces nouns and adjectives with antonyms using WordNet [9], and replaces named entities and numbers with the closest words in the GloVe word vector space [10]. A fake answer is generated using NER and POS tags from Stanford CoreNLP [11], and then the question is transformed into a statement using some predefined rules, after which crowdsourcing is done for grammar checking. The ultimate result is a sentence that is syntactically similar to the question, which is then appended to the end of the original paragraph (or the beginning). This method has been shown to drop accuracy by more than 50% even on state-of-the-art models, during the time that the method was created. The latest state-of-the-art models like BERT [12], are increasingly more robust to these attacks, but still face a drop in accuracy of up to 30%.

AddAny appends a sequence of words to the end of the context paragraph by greedily choosing words that result in decreased softmax probabilities of the model. While AddAny is also very successful, it is not designed to be inconspicuous to a human viewer, and thus, a human reading the context paragraph would be capable of identifying that the paragraph had been modified because the appended material is merely a sequence of words with no sentence structure.

While AddSent is a brilliant method for identifying flawed language understanding by models, the sentence that has been added is not constrained to fit the context paragraph cohesively, and as a result, it often sticks out from the context paragraph as being unrelated to the context, despite being successful at fooling the model. In this paper, we present manual and automated techniques to alter context paragraphs, in order to lower the accuracy scores of models, with the hope that the ideas in these methods could be used to evaluate the extent to which machine comprehension models understand the context paragraph and what is being asked of them by the accompanying questions, and to thus motivate the development of even more powerful and robust NLP models. Furthermore, we examine why these adversarial attacks are so successful and what is lacking in current question answering models.

2 Methodology

The dataset was collected from the SQuAD website [1]. The accuracy is measured using two metrics: Exact Match (EM) and F1 score.

The rationale behind the adversarial modifications to the paragraphs was that the tendency of the models to perform pattern-matching rather than context understanding had to be tackled. For this, several methods were attempted on two models that have achieved high scores on the SQuAD dataset: BERT [12] and R-NET [13]. The reasons for using these models was that both of them were, at one point, the state-of-the-art model on the SQuAD leaderboard, yielding the best score, but they both use a different type of architecture that can be used to highlight the merits and weaknesses of each type of architecture with regards to what depth of understanding the models reach with reference to the machine comprehension task. BERT uses the self-attention based Transformer architecture [14] and R-NET uses a simpler bi-directional GRU-based [15] model with attention.

The methods that were used involved Paragraph Paraphrasing, Question Paraphrasing, and different strategies to add irrelevant information to the paragraphs in order to trick the model into choosing the answer from the appended

[1] https://rajpurkar.github.io/SQuAD-explorer/

sentences, in a manner similar to AddSent, but with some fundamental differences, namely ensuring that the context of the appended sentence matches the existing paragraph.

2.1 Paraphrasing context paragraphs

The context paragraph is paraphrased using both manual and automated methods. The paraphrase could be done on the entire paragraph, or it could be limited to the sentences which are known to contain the answer to the accompanying question.

The standard evaluation metrics of SQuAD, namely EM and F1, are not used to evaluate the performance of the model under this modified dataset because the model answers are not guaranteed to match the original ground truth answers, due to the fact that the words used in the context paragraph might have changed during the paraphrasing.

2.2 Paraphrasing the questions

In this method, the questions are paraphrased manually, so that the syntax of the question is as different as possible from the syntax of the sentence that contains the answer. In addition, wherever possible, the question word (who, what, why, which, where, how) is replaced with another question word, so as to trick the model into looking for some information that corresponds to how the replaced question word is conventionally used. For example, *"Where did Martin Luther go to school?"* can be paraphrased as *"Which place of higher learning did Martin Luther attend?"*.

2.3 Adding irrelevant information to the paragraphs

This method is similar to AddSent in that some sentence that resembles the syntax of the question, is added to the context paragraph. The content of the additional sentence must be devised in such a way that it neither affirms nor contradicts anything that has been stated in the paragraph. However, unlike AddSent which appends sentences that generally have no connection to the paragraph, the sentences added in this method are constrained to be congruous with the context of the paragraph, notwithstanding the fact that the information that is presented within the appended sentences is ultimately irrelevant to the paragraph. For instance, the question *"In the encoding of mathematical objects, what is the way in which integers are commonly expressed?"* can be converted to the statement *"In the encoding of mathematical objects, the way in which integers are commonly expressed is different from how humans express them."*, which is added to the position in the paragraph where it best fits without appearing conspicuously absurd. Generally, this method does not require any domain knowledge, and can be formulated merely by reading the context paragraph and question. The only requirements are that the added sentence is as similar as possible in syntax with the accompanying question, and that the added sentence does not contain the answer, and that it does not contradict the correct answer. There are several ways to do this, and these methods are further elaborated in section 3.3.3.

3 Experiments

3.1 Models

The Tensorflow [16] implementations of BERT and R-Net models were both obtained from Github. The available R-NET model is already trained and can be readily used on SQuAD. This implementation achieves an EM score of 71.07 and F1 score of 79.51 on the official development set. The pre-trained BERT model was available, which had to be fine-tuned on the SQuAD dataset, after which it was ready to use. It had an EM score of 81.25 and F1 score of 88.41 on the official development sets.

3.2 AddSent

The models were tested with AddSent using the official modified dataset by Jia and Liang [1]. The results obtained (Table 1) are fairly consistent with that of the original paper.

3.3 Main Experiments

3.3.1 Paraphrasing context paragraphs

A small sample of context paragraphs from the SQuAD dataset were paraphrased. The sample consisted of 40 paragraphs, 20 of which were manually paraphrased, and 20 of which were paraphrased using the online services QuillBot [17] and Paraphrasing Tool [18], and Paraphrase Online [19]. The manual paraphrases were attempted with

Table 1: The accuracies of the models under the original and adversarially modified datasets of AddSent provided by Robin Jia and Percy Liang.

Model Performance on AddSent				
Model	EM Score on Original Dataset	EM Score on the adversarial AddSent Dataset	F1 Score on Original Datase	F1 Score on the adversarial AddSent Dataset
R-NET	70.60	39.70	78.55	45.81
BERT	80.90	53.60	88.28	59.96

Article: : Martin Luther
Original Paragraph: Luther taught that salvation and subsequently eternal life is not earned by good deeds but is received only as a free gift of God's grace through faith in Jesus Christ as redeemer from sin. His theology challenged the authority and office of the Pope by teaching that the Bible is the only source of divinely revealed knowledge from God and opposed sacerdotalism by considering all baptized Christians to be a holy priesthood. Those who identify with these, and all of Luther's wider teachings, are called Lutherans even though Luther insisted on Christian or Evangelical as the only acceptable names for individuals who professed Christ.

Paraphrased Paragraph: Luther propagated the idea that redemption and afterward perpetual life is not received by good actions but is established only as a gratuitous gift of God's grace through confidence in Jesus Christ as liberator from immorality. His theology refuted the superior status and office of the Pope by instructing people that the only source of divinely revealed knowledge from God is from the Bible, and he was against sacerdotalism by accepting every baptized Christian as a holy priest. People who relate to these principles, and all of Luther's wider pedagogy, are referred to as Lutherans even though Luther maintained that Christian or Evangelical as the only suitable names for people who acknowledged Christ.

Figure 2: Simple paraphrase that does not alter the sentence structure significantly.

differing complexities of paraphrase. Some paraphrases were fairly standard, i.e. changing many words to their synonyms and using slightly different sentence structures (Figure 2), while others were more complicated, changing words to highly obscure synonyms, and rephrasing using different sentence structures, particularly for the sentences that contained the answers to the questions associated with the context paragraph (Figure 3).

The evaluation was done manually, as the EM and F1 scores cannot give an accurate representation of model accuracy if the words used in the answers are different. In order to perform manual evaluation, a simple percentage-based metric was used, which is more lenient than the F1 score. To evaluate the answer, the model answer was analyzed and compared with the ground truth answer. If the answers were synonymous to each other, or if the answers differed, but the model's answer could still be construed as correct given the possibly modified semantics of the context paragraph after the paraphrase, the model's answer is considered correct. The R-NET model faces an approximately 2% decrease in accuracy under this adversarial dataset, while the BERT model only faces a roughly 1% decrease in accuracy. The results were calculated over the questions from both the manually paraphrased paragraphs and the automatically

Article: : Martin Luther
Original Paragraph: Martin Luther's translation of the Bible into the vernacular (instead of Latin) made it more accessible, which had a tremendous impact on the church and German culture. It fostered the development of a standard version of the German language, added several principles to the art of translation, and influenced the writing of an English translation, the Tyndale Bible. His hymns influenced the development of singing in churches. His marriage to Katharina von Bora set a model for the practice of clerical marriage, allowing Protestant clergy to marry.

Paraphrased Paragraph: His translation of the Bible into the local dialects (as opposed to Latin) offered more people with the opportunity to read it, which impacted the Church and German culture tremendously. It was instrumental for the creation of a standard version of German, and augmented the art of translation with several new concepts, and further affected the development of the English translation, which came to be known as the Tyndale Bible. His hymns shaped the development of hymn-chanting in churches. His marriage to Katharina von Bora laid a model for clerical marriage practice, permitting the marriage of Protestant clergy.

Figure 3: A paraphrase that makes some alteration to the sentence structure, while ultimately retaining the original meaning of the paragraph.

paraphrased paragraphs. However, both the manual and automated paraphrase techniques seem to yield similar drops in accuracy.

Both models only face a very slight decrease in accuracy when the context paragraphs are paraphrased, which is expected given that the meanings of the words in the paragraphs should be reasonably captured with their embedding layers. R-NET uses the GLoVe embeddings [10], which are not dependent on context, but nevertheless, it is able to answer most questions correctly and only faces a 2% drop in accuracy in a sample with a mixture of simple and complex paraphrases. BERT on the other hand, is even more robust to this type of attack, which can be attributed to its context-dependent embeddings, as well as its more complex architecture.

Table 2: The percentage accuracies of the models, computed by manually judging whether the model output was acceptable, under the original samples and adversarially modified samples which consist of paraphrased paragraphs.

Model Performance when paraphrasing context paragraphs		
Model	Original samples	Samples with paraphrased context paragraphs
R-NET	77.23	75.05
BERT	85.53	84.38

3.3.2 Paraphrasing questions

The original questions of a small sample of the dataset were paraphrased manually. This sample consisted of 30 context paragraphs, each with 4-5 accompanying questions. Figure 4 shows an example of how these paraphrases were created. The method tried to alter the syntax so that it did not resemble the actual answer, and wherever possible, replace question words, and rephrase the question accordingly to retain its grammatical correctness (as long as the meaning did not change). When tested on this adversarial data The EM score of R-NET drops by about 11 points and its F1 score drops by 12 points. BERT, on the other hand, actually sees an increase in its accuracy. It faces a roughly 0.1 point increase in its EM score and a roughly 0.3 point increase in its F1. Interestingly, in the BERT model, some questions that it had initially answered wrongly, were answered correctly after paraphrasing the question, and overall, it seems quite robust to this kind of adversarial attack. It should be noted, however, that there is no single way to paraphrase a sentence, and that the quality of paraphrasing and its effect on a model will differ depending on the paraphrase. As long as the above-mentioned general guidelines are followed when creating a paraphrase, the model accuracy should drop for models with attention-based architectures that use recurrent units like Gated Recurrent Units (GRUs) [15] or Long Short-Term Memory Units (LSTMs) [20].

The idea behind using this method was to force the model to focus on some other part of the text besides the answer, by changing the structure of the question and changing the interrogative pronoun wherever possible. The reason for its success on R-NET could be attributed to the pattern-matching nature of its simpler attention-based recurrent architecture. This could explain why it differentiates questions like "*How early did Luther say he had to awaken every day?*" and "*When was Luther made to get out of bed every day?*" from Figure 4 and makes the predictions "*four*" and "*1501*" respectively. The word "*when*" is probably forcing it to look for a date. The reason BERT does not face the same vulnerability could be because of its self-attention based architecture, which explains how it is able to identify that questions like "*Which company owns ABC?*" and "*Who owns ABC?*" are asking for the same answer.

Table 3: The accuracies of the models under the original samples and adversarially modified samples with paraphrased questions.

Model Performance when paraphrasing questions				
Model	EM Score on original sample	EM Score on the adversarial sample	F1 Score on original sample	F1 Score on the adversarial sample
R-NET	73.671	62.308	84.718	72.256
BERT	74.117	74.231	84.759	85.084

3.3.3 Addition of irrelevant information into the context paragraph

Besides AddSent, this method was the most effective in lowering model accuracy out of all the methods we tried. The simplistic nature of the questions in the SQuAD dataset are exploited to generate noisy statements within the passage,

Article: : Martin Luther
Original Paragraph: In 1501, at the age of 19, he entered the University of Erfurt, which he later described as a beerhouse and whorehouse. He was made to wake at four every morning for what has been described as "a day of rote learning and often wearying spiritual exercises." He received his master's degree in 1505.

Question: Where did Martin Luther go to school?

Model Prediction: *University of Erfurt*

Question: How did Luther describe the University of Erfurt?

Model Prediction: *as a beerhouse and whorehouse*

Question: How early did Luther say he had to awaken every day?

Model Prediction: *four*

Question: How did Luther describe his learning at the university?

Model Prediction: *a day of rote learning and often wearying spiritual exercises*

Question: In what year did Luther get his degree?

Model Prediction: *1505*

Question: Which place of higher learning did Martin Luther attend?

Model Prediction: *rote*

Question: What was Luther's description of the University of Erfurt?

Model Prediction: *as a beerhouse and whorehouse*

Question: When was Luther made to get out of bed every day?

Model Prediction: *1501*

Question: What was Luther's impression of the learning at the institute of higher learning that he was attending?

Model Prediction: *a day of rote learning and often wearying spiritual exercises*

Question: When was Luther's degree conferred to him?

Model Prediction: *1505*

Figure 4: An example of question paraphrasing, and the result of this example on R-NET's predictions.

which act as distractors for the model, attempting to compel it to choose the answer from an appended sentence. Some patterns of generating questions are more effective than others. We attempted to add irrelevant information to the context paragraphs using two main methods.

Question word substitution patterns Questions that use the word *what* make up almost 50% of the questions in the SQuAD dev set. The pattern that exploits this type of questions involves converting the question to a declarative statement, keeping as much of the original syntax and vocabulary as possible, and substituting the answer with an abstract concept or description as the object of this declarative sentence. For example, in a question about computational complexity theory, the question "*What are the two simple word responses to a decision problem?*" has an answer "*yes or no*". This answer is found in this sentence from the paragraph: "*A decision problem is a special type of computational problem whose answer is either yes or no, or alternately either 1 or 0.*" In order to use it to create an adversarial statement, this question is turned into the statement: "*The two simple word responses to a decision problem are fairly straightforward.*", and this statement is placed before the sentence from the paragraph that contains the answer. Similarly, for other examples, some form of irrelevant information is added using this method and is placed at a position in the paragraph where it would fit best, in terms of readability and in terms of how inconspicuous it would appear to a human reader.

Similarly, for questions asking for *where*, *when*, *who*, *which*, *why*, and *how*, some sort of irrelevant information is added to replace the question word when the question is converted to a declarative sentence. In figures 5 and 6, examples of the kind of irrelevant information that can be used have been presented, along with how sentences are framed with this method, and how these sentences would fit within the paragraph. The general strategy to determine the structure of the appended question is to include something that the model might believe is the answer, based on the question word. For example, if the question begins with *where*, a sentence is added such that the sentence contains a location, and similarly for *when*, *who*, and *which*.

Using simple conditional statements and POS-tagging, this method was automated in a heuristic manner. A pool of irrelevant information was created, and this irrelevant information was randomly sampled during the creation of the adversarial statement. This pool included phrases like "*sometimes easy to dismiss*", "*a debatable concept*", "*a point*

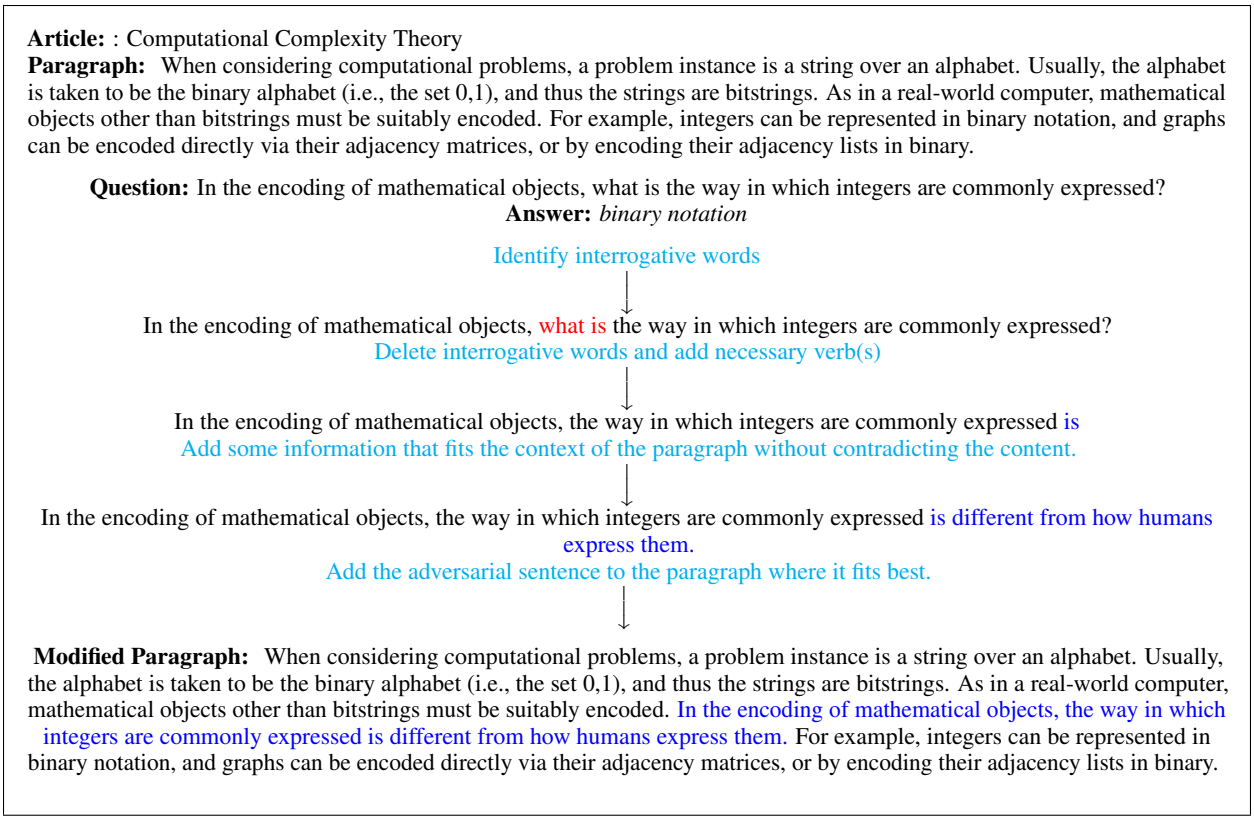

Figure 5: An example of a way to add irrelevant information to the paragraph, by rephrasing the question into a statement with similar syntax, but with different content such that it fits into the paragraph coherently, without altering the meaning of the paragraph.

that is important to discuss", and so on. For simplicity, this automated method only consider questions that begin with the words *what*, *who*, *which*, *when*, and *where*, and ignores all other questions. It should be noted that even if the question has one of the above-mentioned question words as the primary interrogative pronoun but does not begin with the question word, then it is still ignored. For instance, this question would be ignored when generating an adversarial statement: "*In the United States the UMC ranks as the largest what?*" Similarly, a pool of random dates and other words corresponding to the word *when* were created. Some examples include "*1911*", "*that summer*", and "*July*". To generate words for the word *where*, names of countries were generated at random using PyPi's country-list package. For questions beginning with *who*, a random named entity from the context paragraph was chosen, and a sentence was framed using the same syntax and vocabulary from the question, using a pool of distractor clauses, like "*thought about*", "*contemplated*", and so on, such that the added sentence did not contradict the information in the paragraph. For example, the question "*Who was this season's NFL MVP?*" would be converted to "*Denver was interested in this season's NFL MVP.*", where "*Denver*" is a random named entity from the context paragraph, "*interested in*" is from the pool of distractor clauses, and the rest of the sentence is from the question itself. In the automated method, the generated sentence was added into the context paragraph either before or after the sentence that contains the answer to the accompanying question using a heuristic method that chooses the position based on which placement position was located closest to the answer in the sentence, but random placement before or after the sentence could also have been used instead.

Using this automated method resulted in a lot of additional sentences within each paragraph, which affected the readability of the passage by humans, due to the fact that most questions began with the above-mentioned question words, and that each paragraph had four to five questions. This often led to paragraphs like the one in Figure 7, which is increasingly more confusing and ungrammatical than the original paragraph but is still answerable by human evaluation. Furthermore, the generated adversarial dataset was not strictly constrained to be grammatically correct. Some heuristics were used to keep the majority of the sentences roughly grammatically correct. When tested on a sample of 4070 paragraphs, where each question that is not ignored contributes to an adversarial sentence in the paragraph, the EM and F1 accuracy of BERT drop by about 12 points, while that of R-NET drops by about 11 points.

Article: : Computational Complexity Theory
Paragraph: When considering computational problems, a problem instance is a string over an alphabet. The name of the alphabet that is most commonly used in a problem instance is related to a different base of numbering systems. Usually, the alphabet is taken to be the binary alphabet (i.e., the set 0,1), and thus the strings are bitstrings. As in a real-world computer, mathematical objects other than bitstrings must be suitably encoded. In the encoding of mathematical objects, the way in which integers are commonly expressed is different from how humans express them. For example, integers can be represented in binary notation, and graphs can be encoded directly via their adjacency matrices, or by encoding their adjacency lists in binary. One way in which graphs can be encoded is unintuitive.

Question: What is the name of the alphabet that is most commonly used in a problem instance?
Original Prediction: *binary alphabet*
Prediction under adversary: *related to a different base of numbering systems*

Question: In the encoding of mathematical objects, what is the way in which integers are commonly expressed?
Original Prediction: *binary notation*
Prediction under adversary: *different*

Question: What is one way in which graphs can be encoded?
Original Prediction: *adjacency matrices*
Prediction under adversary: *unintuitive*

Figure 6: An example that demonstrates a way to add irrelevant information to the paragraph, by rephrasing the question into a statement with similar syntax, but with different content such that it fits into the paragraph coherently, without altering the meaning of the paragraph. The figure shows a comparison between the original prediction by the BERT model and its prediction under the adversary.

Both models are fairly robust to this adversarial dataset. They are able to identify what to search for in the paragraph despite the presence of the distractor sentences, most likely due to the attention mechanisms used in both the models. The reason for the success of some of the examples in this dataset on both the models could be due to their the overly stable nature of the models.

A modification that would improve readability and cohesion of the passage would be to only use one question per paragraph to add a sentence to the passage. However, when this method was attempted on the same 4070 paragraphs, the EM and F1 accuracy of BERT dropped only by 1.4%. The most likely reason for this is that the question to be converted to a statement was chosen randomly from the list of accompanying questions for each paragraph. If this method was modified to greedily choose questions that result in a drop in accuracy, similar to the way the AddSent adversarial dataset was created, it should be able to decrease the accuracy to a greater extent than 1.4%. It should be noted that in the official AddSent implementation, only one question was used to create the distractor sentence, and in the evaluation of the results, all the other questions were deleted. If all the original questions were retained in the dev set, the accuracy drop would be significantly lower.

To solve the cohesion and readability issues, this method can also be performed manually, which results in far more readable paragraphs, because the adversarial sentences can be adjusted based on context. When tested on a small sample of paragraphs, the manual implementation of this method resulted in a 13 point decrease in EM and F1 scores on the BERT model, which is a slightly larger drop than that of the automated method. In this method, the adversarial sentences appended to the paragraph are much more inconspicuous to the human reader compared to the automated method, which makes it a more effective kind of adversarial attack.

Table 4: The accuracies of the models under the original samples and adversarially modified samples with the automated method that uses the question words substitution patterns.

Model Performance when adding irrelevant information to the context paragraph (automated)				
Model	EM Score on original sample	EM Score on the adversarial sample	F1 Score on original sample	F1 Score on the adversarial sample
R-NET	70.60	61.30	78.55	67.71
BERT	80.90	68.96	88.28	76.78

AddExtraneous: A Context-based addition of irrelevant information Our AddExtraneous method is inspired by the AddSent method by Jia and Liang. This method uses the context of the information in the paragraph to add some

Article: : Super Bowl 50
Paragraph: The seed that was the Carolina Panthers was well-known. The seed that was the Denver Broncos was debatable. For the third straight season, the number one seeds from both conferences met in the Super Bowl. The team in super bowl 50 that had a 15-1 record are difficult to comprehend. The Carolina Panthers became one of only ten teams to have completed a regular season with only one loss, and one of only six teams to have acquired a 15–1 record, while the Denver Broncos became one of four teams to have made eight appearances in the Super Bowl. The number of times the Denver Broncos played in a super bowl by the time they reached super bowl 50 was fairly straightforward. The last super bowl the broncos participated in was important to note. The Broncos made their second Super Bowl appearance in three years, having reached Super Bowl XLVIII, while the Panthers made their second Super Bowl appearance in franchise history, their other appearance being Super Bowl XXXVIII. The first super bowl that the carolina panthers played in was different depending on who is asked. Coincidentally, both teams were coached by John Fox in their last Super Bowl appearance prior to Super Bowl 50.

Question: Who coached each Super Bowl 50 participant in their most recent Super Bowl appearance prior to Super Bowl 50?
Original Prediction: *John Fox*
Prediction under adversary: *John Fox*

Question: Which team in Super Bowl 50 had a 15-1 record?
Original Prediction: *Carolina Panthers*
Prediction under adversary: *Carolina Panthers*

Question: What was the last Super Bowl the Broncos participated in?
Original Prediction: *Super Bowl XLVIII*
Prediction under adversary: *Super Bowl XLVIII*

Question: What seed was the Carolina Panthers?
Original Prediction: *number one*
Prediction under adversary: *The seed that was the Denver Broncos*

Question: What seed was the Denver Broncos?
Original Prediction: *number one*
Prediction under adversary: *Carolina Panthers was well-known*

Figure 7: An example of a heuristic, automated method to add irrelevant information to the paragraph, by rephrasing the question into a statement with similar syntax, but with different context, by placing it at a location close to where the original answer was found. The figure shows a comparison between the original prediction by the BERT model and its prediction under the adversary on five out of eighteen questions associated with this context paragraph. The automated method makes some minor grammatical errors, and the drop in general coherency of the passage makes the added sentences quite conspicuous from the rest of the context paragraph.

irrelevant information that targets a specific question, in a manner that reuses the vocabulary in the question, without contradicting the paragraph. The added sentence does not necessarily need to have similar syntax to the question, but as far as possible, it attempts to retain the original syntax. It is appended to a location in the paragraph where it would fit the most, and in some cases, it does not even have to be a separate sentence, but rather an extension of an existing sentence. Since this method is context-based, it is performed manually.

To perform this method, a question from the paragraph is transformed into a statement using a variety of sentence structures. The transformed statement is made to have an incorrect answer, such that the modified statement directly contradicts what has been stated in the paragraph. This modified sentence is then qualified using some distractor clauses like "*thought about*", "*was going to*", and so on. Then, depending on whether the sentence still contradicts the context paragraph or if the sentence is ambiguous, some clause is added to the end of the statement, such that it nullifies the verb within the part of the sentence that used to contain the contradiction, without the use of the word ″not" as far as possible. The resulting sentence thus does not contradict the context paragraph and should fit into the paragraph cohesively. This method is demonstrated in Figures 8, 9, 10, and 11. Although the method requires some creativity in terms of style of sentence generation, no prior knowledge of the context domain is required. Furthermore, for questions that are very simple in their structure, or for questions that cannot be transformed without introducing absurdity, the question word substitution patterns can be used manually.

This method has proven to be very effective in reducing the accuracy of both the BERT and R-NET models. On a sample of 40 paragraphs, this method resulted in a roughly 28 point and 35 point drop in EM and F1 scores respectively for BERT, and a roughly 32 point and 34 point drop in EM and F1 accuracy respectively for R-NET.

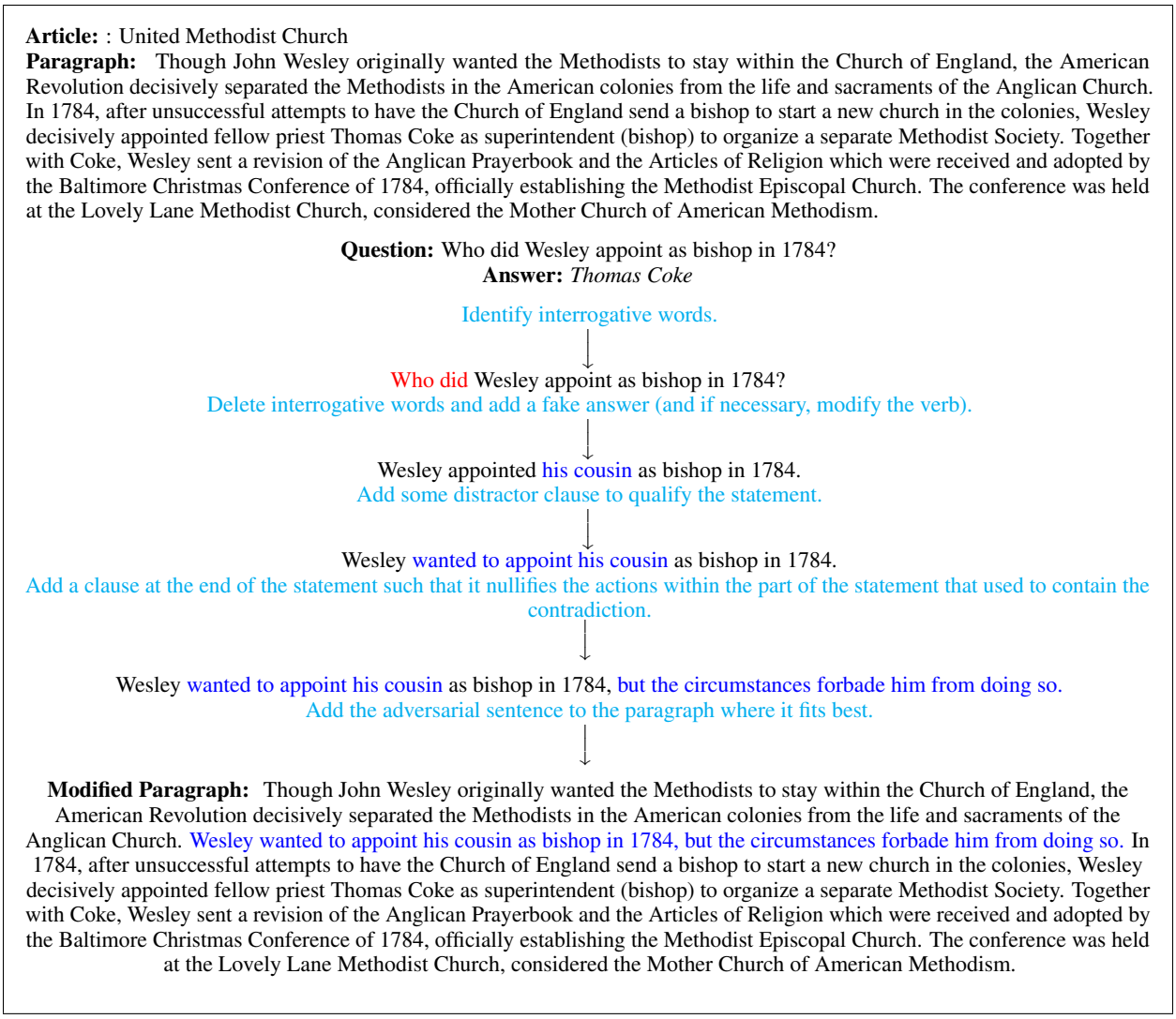

Figure 8: An example of the AddExtraneous method.

AddExtraneous is very effective against both R-NET and BERT because it, like AddSent, exploits the difficulty that these models face with differentiating between referring to (or discussing) a fact and stating the fact outright within a sentence. This explains why they are unable to answer the questions in Figures 9, 10 and 11. For instance, the models misconstrue the sentence "*Wesley wanted to appoint his cousin as bishop in 1784, but the circumstances forbade him from doing so.*", from Figure 10, as having stated that Wesley had appointed his cousin as bishop. The models are not able to understand the connotations of "*but the circumstances forbade him from doing so*", which explains why it picks the answer from the phrase preceding it. For many of the examples that were tested, the models seem to have trouble with choosing an answer from among the sentence that states the fact that has been asked for in the question, and the sentence that talks about what has been asked for in the question. It appears that in these situations, the models often choose the sentence that has more words in common with the question or the sentence that has a syntax closer to the question. For example, in Figure 11, to answer the question "*Where did Luther go to school?*", the correct answer is within the sentence "*In 1501, at the age of 19, he entered the University of Erfurt, which he later described as a beerhouse and whorehouse.*", but the models choose the answer from the sentence "*Martin Luther wanted to go to Harvard University.*" As explained above, the models seem to be unable to understand the connotations of the phrase "*wanted to*", and thus choose the answer "*Harvard University*" from the sentence, presumably because the sentence is syntactically closer to the original question.

Article: : French and Indian War
Paragraph: In Europe, the North American theater of the Seven Years' War usually is not given a separate name. The entire international conflict is known as the Seven Years' War. "Seven Years" refers to events in Europe, from the official declaration of war in 1756 to the signing of the peace treaty in 1763. These dates do not correspond with the fighting on mainland North America, where the fighting between the two colonial powers was largely concluded in six years, from the Battle of Jumonville Glen in 1754 to the capture of Montreal in 1760. Montreal was to be captured in 1759, but due to the opposition, it took an extra year. Despite contrary belief, the fighting in the Seven Years' War was not waged for seven years.

Question: How long did the fighting last in Seven Years War?
Original Prediction: *six years*
Prediction under adversary: *seven years*

Question: When was Montreal captured?
Original Prediction: *1760*
Prediction under adversary: *1759*

Figure 9: An example of the type of adversarial modifications that are made using the AddExtraneous method. The figure shows a comparison between the original prediction made by the BERT model, and its prediction under the adversary.

Article: : United Methodist Church
Paragraph: Though John Wesley originally wanted the Methodists to stay within the Church of England, the American Revolution decisively separated the Methodists in the American colonies from the life and sacraments of the Anglican Church. Wesley wanted to appoint his cousin as bishop in 1784, but the circumstances forbade him from doing so. In 1784, after unsuccessful attempts to have the Church of England send a bishop to start a new church in the colonies, Wesley decisively appointed fellow priest Thomas Coke as superintendent (bishop) to organize a separate Methodist Society. Together with Coke, Wesley sent a revision of the Anglican Prayerbook and the Articles of Religion which were received and adopted by the Baltimore Christmas Conference of 1784, officially establishing the Methodist Episcopal Church. The conference was held at the Lovely Lane Methodist Church, considered the Mother Church of American Methodism.

Question: Who did Wesley appoint as bishop in 1784?
Original Prediction: *Thomas Coke*
Prediction under adversary: *his cousin*

Figure 10: An example of the type of adversarial modifications that are made using the AddExtraneous method. The figure shows a comparison between the original prediction made by the BERT model, and its prediction under the adversary.

This sort of overly stable nature seems to be a problem with many other NLP models including commercial virtual assistants like Sir and Google Assistant. Currently this seems to be one of the hurdles that is preventing NLP models from dealing with language in a more human-like manner.

Table 5: The accuracies of the models under the original samples and adversarially modified samples using the AddExtraneous method of adding irrelevant information to the context paragraph.

Model Performance when adding irrelevant information to the context paragraph using AddExtraneous (manual)				
Model	EM Score on original sample	EM Score on the adversarial sample	F1 Score on original sample	F1 Score on the adversarial sample
R-NET	68.213	36.167	79.626	43.929
BERT	73.592	45.750	87.321	52.018

3.4 Adversarial Training

Adversarial Training was not performed since the manually generated adversarial datasets are too small. The automatically generated dataset from the question word substitution method was not used for adversarial training because the pools of irrelevant data that were added were finite and limited, given that they were created to handle examples with no finetuning for the context of the paragraph. They would most likely have resulted in the model merely ignoring

Article: : Martin Luther
Paragraph: Martin Luther wanted to go to Harvard University. In 1501, at the age of 19, he entered the University of Erfurt, which he later described as a beerhouse and whorehouse. He was made to wake at four every morning for what has been described as "a day of rote learning and often wearying spiritual exercises." He received his Master's degree in 1505.

Question: Where did Martin Luther go to school?
Original Prediction: *University of Erfurt*
Prediction under adversary: *Harvard University*

Figure 11: An example of the type of adversarial modifications that are made using the AddExtraneous method. The figure shows a comparison between the original prediction made by the BERT model, and its prediction under the adversary.

the phrases from the finite pools of irrelevant data, and thus, would not be able to handle more difficult adversarial examples like the ones created by AddAny or AddExtraneous.

4 Related Work

The oversensitive nature of Computer Vision models was exploited by Goodfellow et al [5], using the Fast Gradient Sign Method (FGSM). This involved adding generated noise to the images to perturb the image, resulting in an incorrect prediction of the model.

In the field of Natural Language Processing, a variant of FGSM was explored by Papernot et al [21], which adjusted the method to handle the discrete nature of text data. It successfully performed adversarial attacks on a sentiment analysis task by replacing words from movie reviews with other words from the word embedding space, such that the signed difference of the word embedding of the original word and the word embedding of the replaced word was closest to the model's Jacobian tensor with respect to the original word embedding, in terms of Euclidean distance.

With regards to Machine Reading Comprehension, Jia and Liang [1] presented the AddSent and AddAny methods (among other methods), the details of which, have been explained earlier. Wang and Bansal [3] presented AddSentDiverse, a modification of the AddSent algorithm, that randomizes sentence placement within the context paragraph.

More general strategies for adversarial example generation have been surveyed by Belinkov and Glass [6], Wang et al [7] and Zhang et al [8].

References

[1] Robin Jia and Percy Liang. Adversarial examples for evaluating reading comprehension systems. *CoRR*, abs/1707.07328, 2017.

[2] Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. Squad: 100, 000+ questions for machine comprehension of text. *CoRR*, abs/1606.05250, 2016.

[3] Yicheng Wang and Mohit Bansal. Robust machine comprehension models via adversarial training. *CoRR*, abs/1804.06473, 2018.

[4] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. Intriguing properties of neural networks. In *International Conference on Learning Representations*, 2014.

[5] Ian Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. In *International Conference on Learning Representations*, 2015.

[6] Yonatan Belinkov and James R. Glass. Analysis methods in neural language processing: A survey. *CoRR*, abs/1812.08951, 2018.

[7] Wenqi Wang, Benxiao Tang, Run Wang, Lina Wang, and Aoshuang Ye. A survey on adversarial attacks and defenses in text. *CoRR*, abs/1902.07285, 2019.

[8] Wei Emma Zhang, Quan Z. Sheng, and Ahoud Abdulrahmn F. Alhazmi. Generating textual adversarial examples for deep learning models: A survey. *CoRR*, abs/1901.06796, 2019.

[9] Christiane Fellbaum, editor. *WordNet: an electronic lexical database*. MIT Press, 1998.

[10] Jeffrey Pennington, Richard Socher, and Christopher D. Manning. Glove: Global vectors for word representation. In *Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, 2014.

[11] Christopher D. Manning, Mihai Surdeanu, John Bauer, Jenny Finkel, Steven J. Bethard, and David McClosky. The Stanford CoreNLP natural language processing toolkit. In *Association for Computational Linguistics (ACL) System Demonstrations*, pages 55–60, 2014.

[12] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

[13] Natural Language Computing Group. R-net: Machine reading comprehension with self-matching networks. May 2017.

[14] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. 2017.

[15] Kyunghyun Cho, Bart van Merrienboer, Çaglar Gülçehre, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using RNN encoder-decoder for statistical machine translation. *CoRR*, abs/1406.1078, 2014.

[16] Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Yangqing Jia, Rafal Jozefowicz, Lukasz Kaiser, Manjunath Kudlur, Josh Levenberg, Dan Mané, Rajat Monga, Sherry Moore, Derek Murray, Chris Olah, Mike Schuster, Jonathon Shlens, Benoit Steiner, Ilya Sutskever, Kunal Talwar, Paul Tucker, Vincent Vanhoucke, Vijay Vasudevan, Fernanda Viégas, Oriol Vinyals, Pete Warden, Martin Wattenberg, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. TensorFlow: Large-scale machine learning on heterogeneous systems, 2015. Software available from tensorflow.org.

[17] Quillbot. `https://quillbot.com/`, 2019. [Online; accessed 16-August-2019].

[18] Paraphrasing tool. `https:///paraphrasing-tool.com/`, 2019. [Online; accessed 16-August-2019].

[19] Paraphrasing-online.com. `https://www.paraphrase-online.com/`, 2019. [Online; accessed 16-August-2019].

[20] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural Comput.*, 9(8):1735–1780, November 1997.

[21] Nicolas Papernot, Patrick D. McDaniel, Ananthram Swami, and Richard E. Harang. Crafting adversarial input sequences for recurrent neural networks. *CoRR*, abs/1604.08275, 2016.