Evaluating Neural Model Robustness for Machine Comprehension

Winston Wu
Center for Language and Speech Processing
Department of Computer Science
Johns Hopkins University
wswu@jhu.edu

Dustin Arendt[1] **Svitlana Volkova**[2]
[1]Visual Analytics Group
[2]Data Sciences and Analytics Group
Pacific Northwest National Laboratory
first.last@pnnl.gov

Abstract

We evaluate neural model robustness to adversarial attacks using different types of linguistic unit perturbations – character and word, and propose a new method for strategic sentence-level perturbations. We experiment with different amounts of perturbations to examine model confidence and misclassification rate, and contrast model performance with different embeddings BERT and ELMo on two benchmark datasets SQuAD and TriviaQA. We demonstrate how to improve model performance during an adversarial attack by using ensembles. Finally, we analyze factors that affect model behavior under adversarial attack, and develop a new model to predict errors during attacks. Our novel findings reveal that (a) unlike BERT, models that use ELMo embeddings are more susceptible to adversarial attacks, (b) unlike word and paraphrase, character perturbations affect the model the most but are most easily compensated for by adversarial training, (c) word perturbations lead to more high-confidence misclassifications compared to sentence- and character-level perturbations, (d) the type of question and model answer length (the longer the answer the more likely it is to be incorrect) is the most predictive of model errors in adversarial setting, and (e) conclusions about model behavior are dataset-specific.

1 Introduction

Deep neural models have recently gained popularity, leading to significant improvements in many Natural Language Understanding (NLU) tasks (Goldberg, 2017). However, the research community still lacks in-depth understanding of how these models work and what kind of linguistic information is actually captured by neural networks (Feng et al., 2018). Evaluating model robustness to manipulated inputs and analyzing model behavior during adversarial attacks can provide deeper insights into how much language understanding models actually have (Hsieh et al., 2019; Si et al., 2020). Moreover, as has been widely discussed, models should be optimized not only for accuracy but also for other important criteria such as reliability, accountability and interpretability (Lipton, 2018; Doshi-Velez and Kim, 2017; Ribeiro et al., 2016; Goodman and Flaxman, 2017).

Context: One of the most famous people born in Warsaw was Maria Skłodowska-Curie, who achieved international recognition for her research on radioactivity and was the first female recipient of the Nobel Prize.
Question: What was Maria Curie the first female recipient of?
Answer: Nobel Prize

Table 1: Example MC question from SQuAD.

In this work, we evaluate neural model robustness on machine comprehension (MC), a task designed to measure a system's understanding of text. In this task, given a *context* paragraph and a *question*, the machine is tasked to provide an *answer*. We focus on *span-based* MC, where the model selects a single contiguous span of tokens in the context as the answer (Tab. 1). We (1) quantitatively measure when and how the model is robust to manipulated inputs, when it generalizes well, and when it is less susceptible to adversarial attacks, (2) demonstrate that relying on ensemble models increases robustness, and (3) develop a new model to predict model errors during attacks. Our novel contributions shed light on the following questions:

- Which embeddings are more susceptible to noise and adversarial attacks?
- What types of text perturbation lead to the most high-confidence misclassifications?
- How does the amount of text perturbation effect model behavior?
- What factors explain model behavior under perturbation?
- Are the above dataset-specific?

Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics, pages 2470–2481
April 19 - 23, 2021.

Broader Implications We would like to stress the importance of this type of work to ensure diversity and progress for the computational linguistics community. We as a community know how to build new models for language understanding, but we do not fully understand how these models work. When we deploy these models in production, they fail to perform well in real-world conditions, and we fail to explain why they fail; the reason being we have not performed through evaluation of model performance under different experimental conditions. Neural model evaluation and thorough error analysis, especially for tasks like machine comprehension, are critical to make progress in the field. We have to ensure our research community goes beyond F1 scores and incremental improvements and gains deeper understanding of models decision making processes to drive revolutionary research rather than evolutionary.

2 Background

There is much recent work on adversarial NLP, surveyed in Belinkov and Glass (2019) and Zhang et al. (2019). To situate our work, we review relevant research on the black-box adversarial setting, in which one does not have access or information about the model's internals, only the model's output and its confidence about the answer.[1]

In an adversarial setting, the adversary seeks to mislead the model into producing an incorrect output by slightly tweaking the input. Recent work has explored input perturbations at different linguistic levels: character, word, and sentence-level. For *character-level perturbations*, NLP systems generally do not take into account the visual characteristics of characters. Researchers have explored the effects of adding noise by randomizing or swapping characters and examining its effect on machine translation (MT) (Heigold et al., 2018; Belinkov and Bisk, 2018), sentiment analysis and spam detection Gao et al. (2018), and toxic content detection Li et al. (2018). Eger et al. (2019) replaced with similar looking symbols, and developed a system to replace characters with nearest neighbors in visual embedding space. For *word-level perturbations*, Alzantot et al. (2018) used a genetic algorithm to replace words with contextually similar words, evaluating on sentiment analysis and textual entailment. For *sentence-level perturbations*, Iyyer et al. (2018) generated adversarial paraphrases by controlling the syntax of sentences and evaluating on sentiment analysis and textual entailment tasks. Hu et al. (2019) found that augmenting the training data with paraphrases can improve performance on natural language inference, question answering, and MT. Niu and Bansal (2018) use adversarial paraphrases for dialog models.

Other related work includes Zhao et al. (2018); Hsieh et al. (2019), who generated natural looking adversarial examples for image classification, textual entailment, and MT. Specifically for MC, Jia and Liang (2017) added a distractor sentence to the end of the context, Ribeiro et al. (2018) extracted sentence perturbation rules from paraphrases created by translating to and then from a foreign language and then manually judged for semantic equivalence, and (Si et al., 2020) focused on evaluating model robustness for MC.

Unlike earlier work, we empirically show how neural model performance degrades under multiple types of adversarial attacks by varying the amount of perturbation, the type of perturbation, model architecture and embedding type, and the dataset used for evaluation. Moreover, our deep analysis examines factors that can explain neural model behavior under these different types of attacks.

Concurrent with the development of our paper, there has also been a slew of relevant work tackling robustness in neural NLP models, including Adversarial Robustness Toolbox (Nicolae et al., 2018), Advertorch (Ding et al., 2019), Foolbox (Rauber et al., 2020), Advbox (Goodman et al., 2020), OpenAttack (Zeng et al., 2020), TEAPOT (Michel et al., 2019), TextAttack (Morris et al., 2020), TextFooler (Jin et al., 2020), and Robustness Gym (Goel et al., 2021).

3 Methods

We perform comprehensive model evaluation for machine comprehension over several dimensions: the amount of perturbation, perturbation type, model and embedding variation, and datasets.

3.1 Perturbation Type

We examine how changes to the context paragraph (excluding the answer span) affect the model's performance using the following perturbations:

- **Character-level**. In computer security, this is known as a homograph attack. These attacks have been investigated to identify phishing

[1] For other settings (e.g. white-box), we refer the reader to the above surveys.

Original	The connection between macroscopic nonconservative forces and microscopic conservative forces is described by detailed treatment with statistical mechanics.
Character	**T**h**e** **c**onn**ec**t**io**n b**e**twe**e**n macr**o**scopi**c** nonc**o**nservat**i**v**e** force**s** an**d** microsco**pic** con**s**ervative f**o**r**c**es is des**c**rib**e**d by detailed tre**a**tm**e**nt with st**a**t**i**st**i**cal m**e**ch**a**nic**s**.
Word	The connection between macroscopic nonconservative forces and **insects** conservative **troops** is **referred** by detailed treatment with statistical mechanics.
Sentence	The **link** between macroscopic **non-conservative** forces and microscopic conservative forces is described **in detail by** statistical mechanics.

Table 2: Examples of character, word and sentence-level perturbations (bold indicates perturbed text).

and spam (Fu et al., 2006b,a; Liu and Stamm, 2007) but to our knowledge have not been applied in the NLP domain. We replace 25% of characters in the context paragraph with deceptive Unicode characters[2] that to a human are indistinguishable from the original.

- **Word-level**. We randomly replace 25% of the words in the context paragraph with their nearest neighbor in the GLoVe (Pennington et al., 2014) embedding space.[3]
- **Sentence-level**. We use Improved ParaBank Rewriter (Hu et al., 2019), a machine translation approach for sentence paraphrasing, to paraphrase sentences in the context paragraph. We perform sentence tokenization, paraphrase each sentence with the paraphraser, then recombine the sentences.

For character and word perturbations, we use 25% as this is where the performance curve in Heigold et al. (2018) flattens out.[4] Regardless of the type of perturbation, we do not perturb the context that contains the answer span, so that the answer can always be found in the context unperturbed. Because paraphrasing is per sentence, we only modify sentences that do not contain the answer span. An example of each perturbation type is shown in Tab. 2.

3.2 Amount of Perturbation

For each perturbation type, we experiment with perturbing the training data at differing amounts. All models are tested on fully perturbed test data.

- **None**: clean training data.
- **Half**: perturb half the training examples.
- **Full**: perturb the entire train set.
- **Both**: append the entire perturbed data to the entire clean data.[5]
- **Ens**: ensemble model that relies on none, half and full perturbed data; we rely on ensemble voting and only include the word in the predicted answer if any two models agree.

3.3 Model Architecture and Embeddings

BiDAF model with ELMo (Seo et al., 2017; Peters et al., 2018). ELMo is a deep, contextualized, character-based word embedding method using a bidirectional language model. The Bi-Directional Attention Flow model is a hierarchical model with embeddings at multiple levels of granularity: character, word, and paragraph. We use pre-trained ELMo embeddings in the BiDAF model implemented in AllenNLP (Gardner et al., 2018).

BERT (Devlin et al., 2019). BERT is another contextualized embedding method that uses Transformers (Vaswani et al., 2017). It is trained to recover masked words in a sentence as well as on a next-sentence prediction task. The output layer of BERT is fed into a fully-connected layer for the span classification task. Pre-trained embeddings can be fine-tuned to a specific task, and we use the Huggingface PyTorch-Transformers package, specifically *bert-large-cased-whole-word-masking-finetuned-squad* model. We fine-tune for two epochs in each experimental settings.

3.4 Benchmark Datasets

We experiment on two benchmark MC datasets:

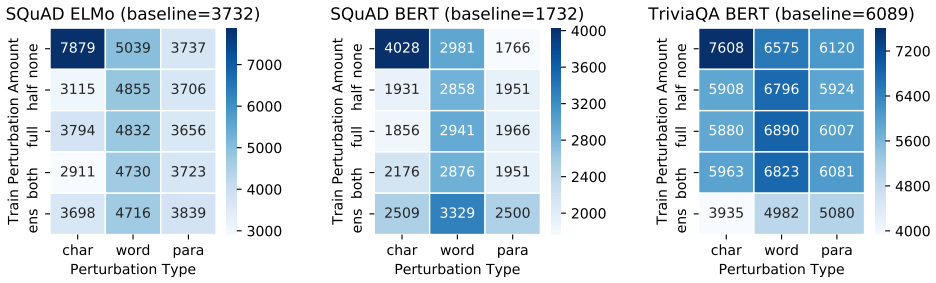
SQuAD (Rajpurkar et al., 2016). The Stanford Question Answering Dataset is a collection of over 100K crowdsourced question and answer pairs. The context containing the answer is taken from Wikipedia articles.

[2]From https://www.unicode.org/Public/security/12.1.0/intentional.txt

[3]Several alternative embedding techniques could be used to find the nearest neighbors e.g., Word2Vec or FastText. We use GLoVe for consistency with previous work (Li et al., 2018).

[4]Belinkov and Bisk (2018) perturbed text at 100% while Heigold et al. (2018) experimented with 5–30% perturbations.

[5]This has twice the amount of data as other settings so is not directly comparable, but many papers show that doing this can improve a model's performance.



(a) SQuAD, ELMo (b) SQuAD, BERT (c) TriviaQA, BERT

Figure 1: Number of errors by perturbation type and amount of perturbation (higher = worse model performance, or more successful attacks). *Baseline* indicates model errors whose training and testing data were not perturbed. For cross-model/embedding comparison, compare (a) and (b). For cross-dataset comparison, compare (a) and (c). The *ens* training setting is an ensemble of results from the none, half, and full settings.

TriviaQA (Joshi et al., 2017). A collection of over 650K crowdsourced question and answer pairs, where the context is from web data or Wikipedia. The construction of the dataset differs from SQuAD in that question answer pairs were first constructed, then evidence was found to support the answer. We utilize the Wikipedia portion of TriviaQA, whose size is comparable to SQuAD. To match the span-based setting of SQuAD, we convert TriviaQA to the SQuAD format using the scripts in the official repo and remove answers without evidence.

4 Evaluation Results

Fig. 1 summarizes our findings on how model behavior changes under noisy perturbations and adversarial attacks. Here, we briefly discuss how perturbation type, perturbation amount, model, and embeddings affect model misclassification rate. In addition, we contrast model performance across datasets and report how to mitigate model error rate using ensembling. Detailed analyses are presented in Sec. 5. Key findings are *italicized*.

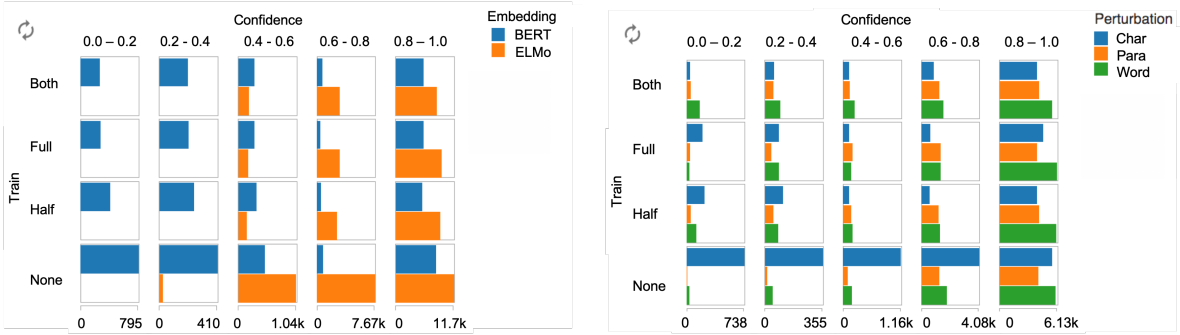
The effect of perturbation type To assess whether perturbations changed the meaning, we ran a human study on a random sample of 100 perturbed contexts from SQuAD. We found (as expected) that the two annotators we employed could not distinguish char-perturbed text from the original. For word perturbations, the meaning of the context remained in 65% of cases, but annotators noted that sentences were often ungrammatical. For sentence-level perturbations, the meaning remained in 83% of cases.

For a model trained on clean data, character perturbations affect the model the most, followed by word perturbations, then paraphrases. To a machine, a single character perturbation results in a completely different word; handling this type of noise is important for a machine seeking to beat human performance. Word perturbations are context independent and can make the sentence ungrammatical.[6] Nevertheless, the context's meaning generally remains coherent. Paraphrase perturbations are most ideal because they retain meaning while allowing more drastic phrase and sentence structure modifications. In Sec. 4.2, we present a more successful adversarially targeted paraphrasing approach.

The effect of perturbation amount Perturbed training data improves the model's performance for character perturbations (1st column of Fig. 1a), likely due to the models' ability to handle unseen words: BiDAF with ELMo utilizes character embeddings, while BERT uses word pieces. Our results corroborate Heigold et al. (2018)'s findings (though on a different task) that *without adversarial training, models perform poorly on perturbed test data, but when models are trained on perturbed data, the amount of perturbed training data does not make much difference.* We do not see statistically significant results for word and paraphrase perturbations (2nd and 3rd columns in each heatmap in Fig. 1). We conclude that perturbing 25% of the words and the non-strategic paraphrasing approach were not aggressive enough.

The effect of model and embedding As shown in Fig. 1a and b, the BERT model had less errors than the ELMo-based model regardless of the perturbation type and amount on SQuAD data. While

[6] Future work will address this with language models.



(a) Across models and embedding

(b) Across perturbation types

Figure 2: The effect of perturbation types and embeddings on model behavior measured as high vs. low confidence misclassifications. More robust models should have less high-confidence error or rate (x-axis).

Train	Test	Model Answer
none	char	here”
half	char	Orientalism
full	char	Orientalism
none	word	Orientalism
half	word	behaviourism identities
full	word	The discourse of Orientalism
none	char	Orientalism
half	char	. . . the East as a negative
full	char	Orientalism

Table 3: Example result from response ensembling under the SQuAD ELMo setting. The question is “What was used by the West to justify control over eastern territories?” The answer is “Orientalism”, and in all three settings, the ensemble was correct.

the two models are not directly comparable, our results indicate that the *BERT model is more robust to adversarial attacks compared to ELMo.*

The effect of the data Holding the model constant (Fig. 1b and c), experiments on TriviaQA resulted in more errors than SQuAD regardless of perturbation amount and type, indicating that TriviaQA may be a harder dataset for MC and may contain data bias, discussed below.

4.1 Adversarial Ensembles

Ensemble adversarial training has recently been explored (Tramèr et al., 2018) as a way to ensure robustness of ML models. For each perturbation type, we present results ensembled from the none, half, and full perturbed settings. We tokenize answers from these three models and keep all tokens that appear at least twice as the resulting answer (Tab. 3). Even when all three model answers differ (e.g. in the word perturbation case), ensembling can often reconstruct the correct answer. Nevertheless, we find that this ensembling only helps for TriviaQA, which has an overall higher error rate (bottom row of each figure in Fig. 1).

4.2 Strategic Paraphrasing

We did not observe a large increase in errors with paraphrase perturbations (Fig. 1), perhaps because paraphrasing, unlike the char and word perturbations, is not a deliberate attack on the sentence. Here we experiment with a novel strategic paraphrasing technique that targets specific words in the context and then generates paraphrases that exclude those words. We find the most important words in the context by individually modifying each word and obtaining the model’s prediction and confidence, a process similar to Li et al. (2018). Our modification consists of removing the word and examining its effect on the model prediction. The most important words are those which, when removed, lower the model confidence of a correct answer or increase confidence of an incorrect answer. The Improved ParaBank Rewriter supports constrained decoding, i.e. specifying positive and negative constraints to force the system output to include or exclude certain phrases. We specify the top five important words in the context as negative constraints to generate strategic paraphrases.[7]

We experimented on 1000 instances in the SQuAD dev set as shown in Tab. 4. Our results indicate that *strategic paraphrasing with negative constraints is a successful adversarial attack*, lowering the F1-score from 89.96 to 84.55. Analysis shows that many words in the question are important and thus excluded from the paraphrases. We also notice that paraphrasing can occasionally turn an incorrect prediction into a correct one. Perhaps

[7]The number of constraints does not necessarily indicate the number of words that are changed in the context.

Original Paragraph	Strategic Paraphrase
. . . Veteran receiver Demaryius Thomas led the team with 105 receptions for 1,304 yards and six touchdowns, while Emmanuel Sanders caught 76 passes for 1,135 yards and six scores, while adding another 106 yards returning punts.	. . . The veteran earman Demaryius Thomas was leading a team of 1,304 yards and six touchdowns, while Emmanuel Sanders caught 76 passes for 1,135 yards and six scores while he added another 106 yards of punts back.
Question: Who led the Broncos with 105 receptions? **Answer**: Demaryius Thomas (correct) → Emmanuel Sanders (incorrect)	

Table 4: Example of strategic paraphrasing: red indicates the important words, which were used as negative constraints in the paraphrasing; blue indicates changed words in the paragraph.

paraphrasing makes the context easier to understand by removing distractor terms; we leave this for future investigation.

4.3 Model Confidence

In a black-box setting, model confidence is one of the only indications of the model's inner workings. The models we employed do not provide a single confidence value; AllenNLP gives a probability that each word in the context is the start and end span, while the BERT models only give the probability for the start and end words. We compute the model's confidence using the normalized entropy of the distribution across the context words, where n is the number of context words, and take the mean for both the start and end word: $1 - \frac{H_n(\mathbf{s})+H_n(\mathbf{e})}{2}$, where $\mathbf{s}$ and $\mathbf{e}$ are probability distributions for the start and end words, respectively. Low entropy indicates certainty about the start/end location. Since the BERT models only provide probabilities for the start and end words, we approximate the entropy by assuming a flat distribution, dividing the remaining probability equally across all other words in the context.

Comparing confidence across models (Fig. 2a), the *BERT model has lower confidence for misclassifications*, which is ideal. A model should not be confident about errors. Fig. 2b compares confidence across perturbation type. In the *none* training setting, character perturbations introduce the most uncertainty compared to word or paraphrase perturbations. This is expected, since character perturbations result in unknown words. In the adversarial training, word perturbations lead to the highest number of high-confidence errors. Thus, *to convincingly mislead the model to be highly confident about errors, one should use word perturbations.*

5 Robustness Analysis

Here, we do a deeper dive into why models make errors with noisy input. We investigate data characteristics and their association with model errors by utilizing CrossCheck (Arendt et al., 2020), a novel interactive tool designed for neural model evaluation. Unlike several recently developed tools for analyzing NLP model errors (Agarwal et al., 2014; Wu et al., 2019) and understanding ML model outputs (Lee et al., 2019; Poursabzi-Sangdeh et al., 2018; Hohman et al., 2019), CrossCheck is designed to allow rapid prototyping and cross-model comparison to support experimentation.[8]

5.1 The Effect of Question Type, Question and Context Lengths

We examine if models make more errors on specific types of questions in adversarial training, i.e., some questions could just be easier that others. We first examine **question type**:[9] who, what, which, when, where, why, how, and other. The majority of SQuAD questions are *what* questions, while most TriviaQA questions are *other* questions, perhaps indicating more complex questions (Fig. 4a). We see that models usually choose answers appropriate for the question type; even if they are incorrect, answers to *when* questions will be dates or time word spans, and answers to *how many* questions will be numbers. Fig. 4a presents key findings on differences in model misclassifications between two datasets given specific question types. *On the SQuAD dataset, the model finds certain question types, e.g. when and how, easiest to answer regardless of the perturbation type*. Responses to these questions, which generally expect numeric answers, are not greatly affected by perturbations. For TriviaQA, in general we observe more errors across question types compared to SQuAD, i.e. more errors in what, which and who questions.

[8] To reproduce our findings, we will release the tool and interactive notebooks upon publication.

[9] Computed as the first word of the question. Many *how* questions are *how many* or *how much*, rather than how in the "in what manner" sense.

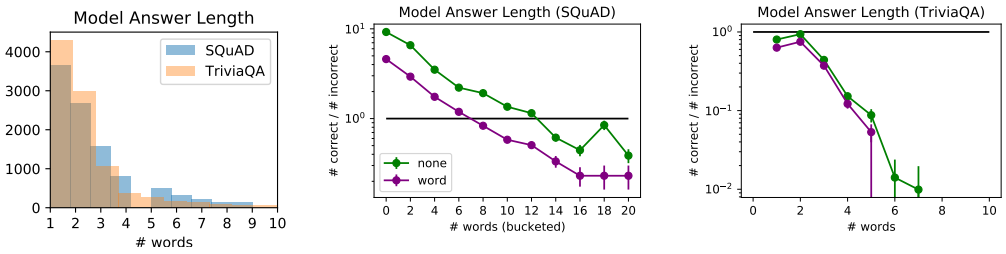
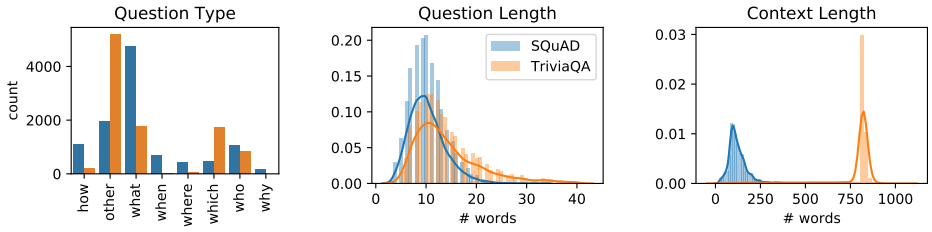
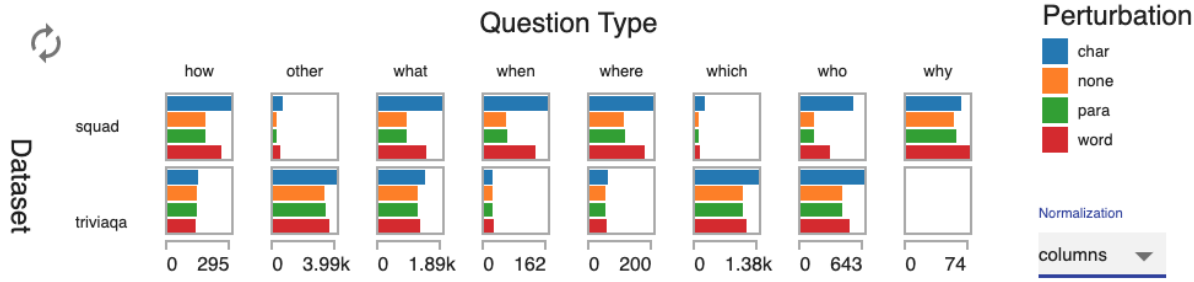


Figure 3: (Left) shows distribution of answer length. (Center) and (Right) show the ratio of correct errors to incorrect errors (log scale y-axis). BERT models were trained on clean data and tested on perturbed data.



(a) Question type distribution (b) Quest. length norm. density (c) Context length norm. density



(d) Error distribution in BERT models across different perturbations and question types

Figure 4: Contrasting MC model errors by question type, and question and context length across datasets.

Regarding **question length**, SQuAD and TriviaQA have similar distributions (Fig. 4b). Both datasets have a mode answer length around 10 words; TriviaQA has a slightly longer tail in the distribution. We did not find question length to impact the error. Regarding **context length**, SQuAD and TriviaQA have vastly differing context length distributions (Fig. 4c), partly due to how the two datasets were constructed (see Sec. 3.4 for details). For both datasets, the error distribution mirrors the context length distribution, and we did not find any relation between model errors and context length.

5.2 The Effect of Answer Length

Our analysis shows that *the length of the model's answer is a strong predictor of model error in the adversarial setting*: the longer the answer length, the more likely it is to be incorrect. Fig. 3 plots the proportion of correct to incorrect answers. We notice a downward trend which is mostly consistent across experimental settings. For both SQuAD and TriviaQA, the models favored shorter answers, which mirrors the data distribution.

5.3 The Effect of Complexity: Annotator Agreement and Reading Level

Here, we examine the effect of task complexity on model performance under adversarial training, using inter-annotator agreement as a proxy for *question* complexity and paragraph readability as a proxy for *context* complexity.

Inter-annotator agreement represents a **question's complexity**: low agreement indicates that annotators did not come to a consensus on the correct answer; thus the question may be difficult to answer. We examine SQuAD, whose questions have one to six annotated answers. In Fig. 5, we present inter-annotator agreement (human confidence) plotted against model confidence over the four training perturbation amounts, looking only at the incorrect predictions. The setting is SQuAD BERT with character perturbation. We observe

Data	Correct	Errors
SQuAD	12.9	13.0
TriviaQA	17.1	17.5

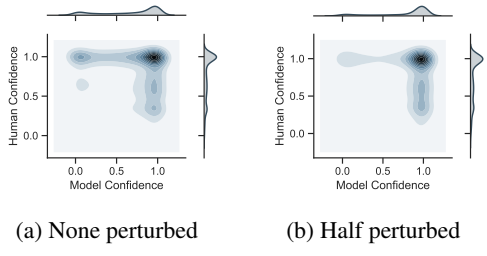
Table 5: Contrasting median readability scores for paragraphs with and without errors across datasets.

that the models are generally confident even when the humans are not, which is noticeable across all perturbation amounts. However, we see interesting differences in model confidence in adversarial training: models trained in the none and half settings have confidence ranging between 0 and 1 compared to the models trained in full and both setting with confidence above 0.8, indicating *training with more perturbed data leads to more confident models*.

To evaluate the effect of **context complexity**, we use the Flesch-Kincaid reading level (Kincaid et al., 1975) to measure readability. For questions the model answered incorrectly, the median readability score was slightly higher than the median score for correct responses (Tab. 5), indicating that *context with higher reading level is harder for the model to understand*. TriviaQA contexts have higher reading levels than SQuAD.

6 Predicting Model Errors

Our in-depth analysis reveals many insights on how and why models make mistakes during adversarial training. Using the characteristics we analyzed above, we developed a binary classification model to predict whether the answer would be an error,



(a) None perturbed (b) Half perturbed

(c) Full perturbed (d) Both

Figure 5: The effect of task complexity on model behavior measured as a joint distribution of errors from BERT model on SQuAD using varied amounts of char perturbations (none, half, full and both).

Embedding	Pert.	Majority	F1 score
ELMo	char	0.58	0.70 ± 0.003
ELMo	word	0.54	0.56 ± 0.004
ELMo	para	0.65	0.65 ± 0.008
BERT	char	0.76	0.77 ± 0.008
BERT	word	0.72	0.73 ± 0.006
BERT	para	0.82	0.82 ± 0.006

Table 6: MC error prediction across datasets, embeddings, and perturbation types.

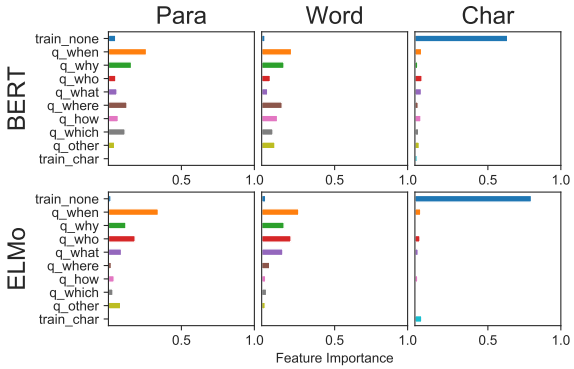


Figure 6: Feature importance when predicting model errors during adversarial attacks.

given the model's answer and attributes of the context paragraph. We one-hot-encode categorical features (training amount, perturbation type, question type) and use other features (question length, context length, answer length, readability) as is. For each setting of embedding and perturbation type on SQuAD, we train an XGBoost model with default settings with 10-fold cross validation (shuffled).

We present the model's average F1 scores (Tab. 6) and feature importance as computed by the XGBoost model (Fig. 6). We see that performance (micro F1) is better to slightly better than a majority baseline (picking the most common class), indicating that certain features are predictive of errors. Specifically, we find that: *for character perturbations, the fact that the training data is clean is a strong predictor of errors; a model trained on clean data is most disrupted by character perturbations; for word and paraphrase perturbations, question types are important predictors of errors.*

7 Conclusion and Future Work

Our in-depth analysis of neural model robustness sheds light on how and why MC models make errors in adversarial training, and through our error prediction model, we discovered features of the data e.g., question types that are strongly predictive of when a model makes errors during adversarial

attacks with noisy inputs. Our results on evaluating the effect of the data e.g., questions and context length will not only explain model performance in context of the data, but will also allow to build future neural models more resilient to adversarial attacks and advance understanding of neural model behavior across a variety of NLU tasks and datasets and its strengths and weaknesses.

For future work, we see many avenues for extension. We plan to experiment with more aggressive and more natural perturbations, and deeper counterfactual evaluation (Pearl, 2019). While recent research has made great strides in increasing model performance on various NLP tasks, it is still not clear what linguistic patterns these neural models are learning, or whether they are learning language at all (Mudrakarta et al., 2018).

More broadly, as AI becomes more entrenched in our lives, AI models need to be held to higher standards including but not limited to accountability (e.g. Wang et al., 2018, 2019, GENIE[10]), fairness (e.g. Saleiro et al., 2018; Bellamy et al., 2018; Bird et al., 2020; Ahn and Lin, 2020, [11];[12];[13]), and transparency (e.g. Lundberg and Lee, 2017; Nori et al., 2019; Hooker et al., 2018; Kokhlikyan et al., 2020; Lundberg et al., 2019; Tenney et al., 2020).

Acknowledgments

The research was performed at Pacific Northwest National Laboratory, a multiprogram national laboratory operated by Battelle for the U.S. Department of Energy. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied of the U.S. Government. The authors would like to thank Maria Glenski for helpful discussions, and the reviewers for their feedback.

References

Apoorv Agarwal, Ankit Agarwal, and Deepak Mittal. 2014. An error analysis tool for natural language processing and applied machine learning. In *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: System Demonstrations*, pages 1–5.

Yongsu Ahn and Y. Lin. 2020. Fairsight: Visual analytics for fairness in decision making. *IEEE Transactions on Visualization and Computer Graphics*, 26:1086–1095.

Moustafa Alzantot, Yash Sharma, Ahmed Elgohary, Bo-Jhang Ho, Mani Srivastava, and Kai-Wei Chang. 2018. Generating natural language adversarial examples. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2890–2896.

Dustin Arendt, Zhuanyi Huang, Prasha Shrestha, Ellyn Ayton, Maria Glenski, and Svitlana Volkova. 2020. Crosscheck: Rapid, reproducible, and interpretable model evaluation.

Yonatan Belinkov and Yonatan Bisk. 2018. Synthetic and natural noise both break neural machine translation. In *6th International Conference on Learning Representations ICLR*.

Yonatan Belinkov and James Glass. 2019. Analysis methods in neural language processing: A survey. *Transactions of the Association for Computational Linguistics*, 7:49–72.

R. Bellamy, K. Dey, M. Hind, Samuel C. Hoffman, S. Houde, K. Kannan, Pranay Lohia, J. Martino, Sameep Mehta, A. Mojsilovic, Seema Nagar, K. Ramamurthy, J. Richards, Diptikalyan Saha, P. Sattigeri, M. Singh, Kush R. Varshney, and Y. Zhang. 2018. Ai fairness 360: An extensible toolkit for detecting, understanding, and mitigating unwanted algorithmic bias. *ArXiv*, abs/1810.01943.

S. Bird, Miro Dudík, R. Edgar, B. Horn, Roman Lutz, Vanessa Milan, M. Sameki, H. Wallach, and Kathleen Walker. 2020. Fairlearn: A toolkit for assessing and improving fairness in ai.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics)*, pages 4171–4186.

G. W. Ding, Luyu Wang, and Xiaomeng Jin. 2019. advertorch v0.1: An adversarial robustness toolbox based on pytorch. *ArXiv*, abs/1902.07623.

Finale Doshi-Velez and Been Kim. 2017. Towards a rigorous science of interpretable machine learning. *arXiv preprint arXiv:1702.08608*.

Steffen Eger, Gözde Gül Şahin, Andreas Rücklé, Ji-Ung Lee, Claudia Schulz, Mohsen Mesgar, Krishnkant Swarnkar, Edwin Simpson, and Iryna Gurevych. 2019. Text processing like humans do: Visually attacking and shielding NLP systems. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics)*, pages 1634–1647.

[10]GENIE https://genie.apps.allenai.org/

[11]ML Fairness Gym https://github.com/google/ml-fairness-gym

[12]Fairness Indicators https://github.com/tensorflow/fairness-indicators

[13]Scikit-Fairness https://github.com/koaning/scikit-fairness

Shi Feng, Eric Wallace, Alvin Grissom II, Mohit Iyyer, Pedro Rodriguez, and Jordan Boyd-Graber. 2018. Pathologies of neural models make interpretations difficult. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3719–3728.

Anthony Y Fu, Xiaotie Deng, Liu Wenyin, and Greg Little. 2006a. The methodology and an application to fight against unicode attacks. In *Proceedings of the second symposium on Usable privacy and security*, pages 91–101. ACM.

Anthony Y Fu, Wan Zhang, Xiaotie Deng, and Liu Wenyin. 2006b. Safeguard against unicode attacks: generation and applications of uc-simlist. In *WWW*, pages 917–918.

Ji Gao, Jack Lanchantin, Mary Lou Soffa, and Yanjun Qi. 2018. Black-box generation of adversarial text sequences to evade deep learning classifiers. In *2018 IEEE Security and Privacy Workshops (SPW)*, pages 50–56. IEEE.

Matt Gardner, Joel Grus, Mark Neumann, Oyvind Tafjord, Pradeep Dasigi, Nelson F. Liu, Matthew Peters, Michael Schmitz, and Luke Zettlemoyer. 2018. AllenNLP: A deep semantic natural language processing platform. In *Proceedings of Workshop for NLP Open Source Software (NLP-OSS)*, pages 1–6.

Karan Goel, Nazneen Rajani, J. Vig, Samson Tan, J. Wu, Stephan Zheng, Caiming Xiong, Mohit Bansal, and C. Ré. 2021. Robustness gym: Unifying the nlp evaluation landscape. *ArXiv*, abs/2101.04840.

Yoav Goldberg. 2017. Neural network methods for natural language processing. *Synthesis Lectures on Human Language Technologies*, 10(1):1–309.

Bryce Goodman and Seth Flaxman. 2017. European union regulations on algorithmic decision-making and a "right to explanation". *AI Magazine*, 38(3):50–57.

Dou Goodman, Xin Hao, Yang Wang, Yuesheng Wu, Junfeng Xiong, and H. Zhang. 2020. Advbox: a toolbox to generate adversarial examples that fool neural networks. *ArXiv*, abs/2001.05574.

Georg Heigold, Stalin Varanasi, Günter Neumann, and Josef van Genabith. 2018. How robust are character-based word embeddings in tagging and MT against wrod scramlbing or randdm nouse? In *Proceedings of the 13th Conference of the Association for Machine Translation*, pages 68–80.

Fred Hohman, Andrew Head, Rich Caruana, Robert DeLine, and Steven M Drucker. 2019. Gamut: A design probe to understand how data scientists understand machine learning models. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, page 579. ACM.

Sara Hooker, D. Erhan, P. Kindermans, and Been Kim. 2018. Evaluating feature importance estimates. *ArXiv*, abs/1806.10758.

Yu-Lun Hsieh, Minhao Cheng, Da-Cheng Juan, Wei Wei, Wen-Lian Hsu, and Cho-Jui Hsieh. 2019. On the robustness of self-attentive models. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1520–1529.

J. Edward Hu, Huda Khayrallah, Ryan Culkin, Patrick Xia, Tongfei Chen, Matt Post, and Benjamin Van Durme. 2019. Improved lexically constrained decoding for translation and monolingual rewriting. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics*, pages 839–850.

Mohit Iyyer, John Wieting, Kevin Gimpel, and Luke Zettlemoyer. 2018. Adversarial example generation with syntactically controlled paraphrase networks. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1875–1885.

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

Di Jin, Zhijing Jin, Joey Tianyi Zhou, and Peter Szolovits. 2020. Is bert really robust? a strong baseline for natural language attack on text classification and entailment. In *AAAI*.

Mandar Joshi, Eunsol Choi, Daniel Weld, and Luke Zettlemoyer. 2017. TriviaQA: A large scale distantly supervised challenge dataset for reading comprehension. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1601–1611.

J Peter Kincaid, Robert P Fishburne Jr, Richard L Rogers, and Brad S Chissom. 1975. Derivation of new readability formulas (automated readability index, fog count and flesch reading ease formula) for navy enlisted personnel.

Narine Kokhlikyan, Vivek Miglani, M. Martín, E. Wang, B. Alsallakh, J. Reynolds, A. Melnikov, Natalia Kliushkina, C. Araya, Siqi Yan, and Orion Reblitz-Richardson. 2020. Captum: A unified and generic model interpretability library for pytorch. *ArXiv*, abs/2009.07896.

Gyeongbok Lee, Sungdong Kim, and Seung-won Hwang. 2019. Qadiver: Interactive framework for diagnosing qa models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 9861–9862.

Jinfeng Li, Shouling Ji, Tianyu Du, Bao Qin Li, and Ting Wang. 2018. Textbugger: Generating adversarial text against real-world applications. *Network and Distributed System Security Symposium*.

Zachary C. Lipton. 2018. The mythos of model interpretability. *ACM Queue*, 16(3):30.

Changwei Liu and Sid Stamm. 2007. Fighting unicode-obfuscated spam. In *Proceedings of the anti-phishing working groups 2nd annual eCrime researchers summit*, pages 45–59. ACM.

Scott Lundberg and Su-In Lee. 2017. A unified approach to interpreting model predictions. In *NIPS*.

Scott M. Lundberg, G. Erion, Hugh Chen, A. DeGrave, J. Prutkin, B. Nair, R. Katz, J. Himmelfarb, N. Bansal, and Su-In Lee. 2019. Explainable ai for trees: From local explanations to global understanding. *ArXiv*, abs/1905.04610.

Paul Michel, Xian Li, Graham Neubig, and J. Pino. 2019. On evaluation of adversarial perturbations for sequence-to-sequence models. In *NAACL-HLT*.

John X. Morris, Eli Lifland, Jin Yong Yoo, Jake Grigsby, D. Jin, and Yanjun Qi. 2020. Textattack: A framework for adversarial attacks, data augmentation, and adversarial training in nlp. In *EMNLP*.

Pramod Kaushik Mudrakarta, Ankur Taly, Mukund Sundararajan, and Kedar Dhamdhere. 2018. Did the model understand the question? In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1896–1906.

Maria-Irina Nicolae, M. Sinn, Minh-Ngoc Tran, Beat Buesser, Ambrish Rawat, M. Wistuba, Valentina Zantedeschi, Nathalie Baracaldo, B. Chen, Heiko Ludwig, Ian Molloy, and Ben Edwards. 2018. Adversarial robustness toolbox v1.0.0. *arXiv: Learning*.

Tong Niu and Mohit Bansal. 2018. Adversarial over-sensitivity and over-stability strategies for dialogue models. In *Proceedings of the 22nd Conference on Computational Natural Language Learning*, pages 486–496. Association for Computational Linguistics.

H. Nori, S. Jenkins, P. Koch, and R. Caruana. 2019. Interpretml: A unified framework for machine learning interpretability. *ArXiv*, abs/1909.09223.

Judea Pearl. 2019. The seven tools of causal inference, with reflections on machine learning. *Commun. ACM*, 62(3):54–60.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543.

Matthew Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics*, pages 2227–2237.

Forough Poursabzi-Sangdeh, Daniel G Goldstein, Jake M Hofman, Jennifer Wortman Vaughan, and Hanna Wallach. 2018. Manipulating and measuring model interpretability. *arXiv preprint arXiv:1802.07810*.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. SQuAD: 100,000+ questions for machine comprehension of text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2383–2392.

Jonas Rauber, R. S. Zimmermann, M. Bethge, and W. Brendel. 2020. Foolbox native: Fast adversarial attacks to benchmark the robustness of machine learning models in pytorch, tensorflow, and jax. *J. Open Source Softw.*, 5:2607.

Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. 2016. Why should i trust you?: Explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, pages 1135–1144. ACM.

Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. 2018. Semantically equivalent adversarial rules for debugging NLP models. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 856–865. Association for Computational Linguistics.

Pedro Saleiro, Benedict Kuester, Abby Stevens, Ari Anisfeld, Loren Hinkson, J. London, and R. Ghani. 2018. Aequitas: A bias and fairness audit toolkit. *ArXiv*, abs/1811.05577.

Min Joon Seo, Aniruddha Kembhavi, Ali Farhadi, and Hannaneh Hajishirzi. 2017. Bidirectional attention flow for machine comprehension. In *5th International Conference on Learning Representations ICLR*.

Chenglei Si, Ziqing Yang, Yiming Cui, Wentao Ma, Ting Liu, and Shijin Wang. 2020. Benchmarking robustness of machine reading comprehension models. *arXiv preprint arXiv:2004.14004*.

Ian Tenney, James Wexler, Jasmijn Bastings, Tolga Bolukbasi, Andy Coenen, Sebastian Gehrmann, E. Jiang, Mahima Pushkarna, Carey Radebaugh, Emily Reif, and A. Yuan. 2020. The language interpretability tool: Extensible, interactive visualizations and analysis for nlp models. In *EMNLP*.

Florian Tramèr, Alexey Kurakin, Nicolas Papernot, Ian J. Goodfellow, Dan Boneh, and Patrick D. McDaniel. 2018. Ensemble adversarial training: Attacks and defenses. In *6th International Conference on Learning Representations ICLR*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008.

Alex Wang, Yada Pruksachatkun, Nikita Nangia, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R. Bowman. 2019. Superglue: A stickier benchmark for general-purpose language understanding systems. In *NeurIPS*.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R. Bowman. 2018. Glue: A multi-task benchmark and analysis platform for natural language understanding. In *BlackboxNLP@EMNLP*.

Tongshuang Wu, Marco Tulio Ribeiro, Jeffrey Heer, and Daniel S Weld. 2019. Errudite: Scalable, reproducible, and testable error analysis. In *Proceedings of the 57th Conference of the Association for Computational Linguistics*, pages 747–763.

Guoyang Zeng, Fanchao Qi, Qianrui Zhou, T. Zhang, Bairu Hou, Yuan Zang, Zhiyuan Liu, and Maosong Sun. 2020. Openattack: An open-source textual adversarial attack toolkit. *ArXiv*, abs/2009.09191.

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

Zhengli Zhao, Dheeru Dua, and Sameer Singh. 2018. Generating natural adversarial examples. In *6th International Conference on Learning Representations ICLR*.