Eric Wallace, Pedro Rodriguez, Shi Feng, Ikuya Yamada, and **Jordan Boyd-Graber**. **Trick Me If You Can: Human-in-the-loop Generation of Adversarial Question Answering Examples**. *Transactions of the Association for Computational Linguistics*, 2019, 14 pages.

```
      @article{Wallace:Rodriguez:Feng:Yamada:Boyd-Graber-2019,
Title = {Trick Me If You Can: Human-in-the-loop Generation of Adversarial Question Answering Examples},
Author = {Eric Wallace and Pedro Rodriguez and Shi Feng and Ikuya Yamada and Jordan Boyd-Graber},
Booktitle = {Transactions of the Association for Computational Linguistics},
Year = {2019},
Volume = {10},
Url = {http://cs.umd.edu/~jbg//docs/2019_tacl_trick.pdf},
}
```

Links:
- Code [https://github.com/Eric-Wallace/trickme-interface]
- Videos [https://www.youtube.com/watch?list=PLegWUnz91WfsBdgqm4wrwdgtPV-QsndlO&v=5sYXzNE07nM&feature=emb_title]
- Data [https://sites.google.com/view/qanta/resources#h.p_d98FBoSkXVUC]

Contact Jordan Boyd-Graber (jbg@boydgraber.org) for questions about this paper.

Trick Me If You Can: Human-in-the-loop Generation of Adversarial Examples for Question Answering

Eric Wallace
EE and UMIACS
University of Maryland
ewallac2@umiacs.umd.edu

Pedro Rodriguez, Shi Feng
CS, UMIACS
University of Maryland
{pedro, shifeng}
@umiacs.umd.edu

Ikuya Yamada
Studio Ousia
ikuya@ousia.jp

Jordan Boyd-Graber
CS, iSchool, UMIACS, LSC
University of Maryland
jbg@umiacs.umd.edu

Abstract

Adversarial evaluation stress tests a model's understanding of natural language. While past approaches expose superficial patterns, the resulting adversarial examples are limited in complexity and diversity. We propose human-in-the-loop adversarial generation, where human authors are guided to break models. We aid the authors with interpretations of model predictions through an interactive user interface. We apply this generation framework to a question answering task called Quizbowl, where trivia enthusiasts craft adversarial questions. The resulting questions are validated via live human–computer matches: although the questions appear ordinary to humans, they systematically stump neural and information retrieval models. The adversarial questions cover diverse phenomena from multi-hop reasoning to entity type distractors, exposing open challenges in robust question answering.

1 Introduction

Proponents of machine learning claim human parity on tasks like reading comprehension (Yu et al., 2018) and commonsense inference (Devlin et al., 2018). Despite these successes, many evaluations neglect that computers solve NLP tasks in a fundamentally different way than humans.

Models can succeed without developing "true" language understanding, instead learning superficial patterns from crawled (Chen et al., 2016) or manually annotated datasets (Kaushik and Lipton, 2018; Gururangan et al., 2018). Thus, recent work

stress tests models via adversarial evaluation: elucidating a system's capabilities by exploiting its weaknesses (Jia and Liang, 2017; Belinkov and Glass, 2019). Unfortunately, while adversarial evaluation reveals simplistic model failures (Ribeiro et al., 2018; Mudrakarta et al., 2018), exploring more complex failure patterns requires human involvement (Figure 1): automatically modifying natural language examples without invalidating them is difficult. Hence, the diversity of adversarial examples is often severely restricted.

Instead, our human–computer hybrid approach uses human creativity to generate adversarial examples. A user interface presents model interpretations and helps users craft model-breaking examples (Section 3). We apply this to a question answering (QA) task called Quizbowl, where trivia enthusiasts—who write questions for academic competitions—create diverse examples that stump existing QA models.

The adversarially-authored test set is nonetheless as easy as regular questions for humans (Section 4), but the relative accuracy of strong QA models drops as much as 40% (Section 5). We also host live human vs. computer matches, where models typically defeat top human teams, but observe spectacular model failures on adversarial questions.

Analyzing the adversarial edits uncovers phenomena that humans can solve but computers cannot (Section 6), validating that our framework uncovers creative, targeted adversarial edits (Section 7). Our resulting adversarial dataset presents a fun, challenging, and diverse resource for future QA research: a system that masters it will demonstrate more robust language understanding.

Figure 1: Adversarial evaluation in NLP typically focuses on a specific phenomenon (e.g., word replacements) and then generates the corresponding examples (top). Consequently, adversarial examples are limited to the diversity of what the underlying generative model or perturbation rule can produce, and also require downstream human evaluation to ensure validity. Our setup (bottom) instead has human-authored examples, using human–computer collaboration to craft adversarial examples with greater diversity.

2 Adversarial Evaluation for NLP

Adversarial examples (Szegedy et al., 2013) often reveal model failures better than traditional test sets. However, automatic adversarial generation is tricky for NLP (e.g., by replacing words) without changing an example's meaning or invalidating it.

Recent work side-steps this by focusing on simple transformations that preserve meaning. For instance, Ribeiro et al. (2018) generate adversarial perturbations such as replacing *What has → What's*. Other minor perturbations such as typos (Belinkov and Bisk, 2018), adding distractor sentences (Jia and Liang, 2017; Mudrakarta et al., 2018), or character replacements (Ebrahimi et al., 2018) preserve meaning while degrading model performance.

Generative models can discover more adversarial perturbations but require post-hoc human verification of the examples. For example, neural paraphrase or language models can generate syntax modifications (Iyyer et al., 2018), plausible captions (Zellers et al., 2018), or NLI premises (Zhao et al., 2018). These methods improve example-level diversity but mainly target a specific phenomenon, e.g., rewriting question syntax.

Furthermore, existing adversarial perturbations are restricted to sentences—not the paragraph inputs of Quizbowl and other tasks—due to challenges in long-text generation. For instance, syntax paraphrase networks (Iyyer et al., 2018) applied to Quizbowl only yield valid paraphrases 3% of the time (Appendix A).

2.1 Putting a Human in the Loop

Instead, we task human authors with *adversarial writing* of questions: generating examples which break a specific QA system but are still answerable by humans. We expose model predictions and interpretations to question authors, who find question edits that confuse the model.

The user interface makes the adversarial writing process interactive and model-driven, in contrast to adversarial examples written independent of a model (Ettinger et al., 2017). The result is an adversarially-authored dataset that explicitly exposes a model's limitations by design.

Human-in-the-loop generation can replace or aid model-based adversarial generation approaches. Creating interfaces and interpretations is often easier than designing and training generative models for specific domains. In domains where adversarial generation is feasible, human creativity can reveal which tactics automatic approaches can later emulate. Model-based and human-in-the-loop generation approaches can also be combined by training models to mimic human adversarial edit history, using the relative merits of both approaches.

3 Our QA Testbed: Quizbowl

The "gold standard" of academic competitions between universities and high schools is Quizbowl. Unlike QA formats such as Jeopardy! (Ferrucci et al., 2010), Quizbowl questions are designed to be interrupted: questions are read to two competing teams and whoever knows the answer first interrupts the question and "buzzes in".

This style of play requires questions to be structured "pyramidally" (Jose, 2017): questions start with difficult clues and get progressively easier. These questions are carefully crafted to allow the most knowledgeable player to answer first. A question on Paris that begins "this capital of France" would test reaction speed, not knowledge; thus, skilled authors arrange the clues so players will recognize them with increasing probability (Figure 2).

The answers to Quizbowl questions are typically well-known entities. In the QA community (Hirschman and Gaizauskas, 2001), this is called "factoid" QA: the entities come from a relatively closed set of possible answers.

> The protagonist of this opera describes the future day when her lover will arrive on a boat in the aria "Un Bel Di" or "One Beautiful Day". The only baritone role in this opera is the consul Sharpless who reads letters for the protagonist, who has a maid named Suzuki. That protagonist blindfolds her child Sorrow before stabbing herself when her lover B. F. Pinkerton returns with a wife. For 10 points, name this Gia4o Puccini opera about an American lieutenant's affair with the Japanese woman Cio-Cio San.
>
> **Answer**: <u>Madama Butterfly</u>

Figure 2: An example Quizbowl question. The question becomes progressively easier (for humans) to answer later on; thus, more knowledgeable players can answer after hearing fewer clues. Our adversarial writing process ensures that the clues also challenge computers.

3.1 Known Exploits of Quizbowl Questions

Like most QA datasets, Quizbowl questions are written for *humans*. Unfortunately, the heuristics that question authors use to select clues do not always apply to computers. For example, humans are unlikely to memorize every song in every opera by a particular composer. This, however, is trivial for a computer. In particular, a simple QA system easily solves the example in Figure 2 from seeing the reference to "Un Bel Di". Other questions contain uniquely identifying "trigger words" (Harris, 2006). For example, "martensite" only appears in questions on <u>steel</u>. For these examples, a QA system needs to understand no additional information other than an if–then rule.

One might wonder if this means that factoid QA is thus an uninteresting, nearly solved research problem. However, some Quizbowl questions are fiendishly difficult for computers. Many questions have intricate coreference patterns (Guha et al., 2015), require reasoning across multiple types of knowledge, or involve complex wordplay. If we can isolate and generate questions with these difficult phenemona, "simplistic" factoid QA quickly becomes non-trivial.

3.2 Models and Datasets

We conduct two rounds of adversarial writing. In the first, authors attack a traditional Information Retrieval (IR) system. The IR model is the baseline from a NIPS 2017 shared task on Quizbowl (Boyd-Graber et al., 2018) based on ElasticSearch (Gormley and Tong, 2015).

In the second round, authors attack either the IR model or a neural QA model. The neural model is a bidirectional RNN using the gated recurrent unit architecture (Cho et al., 2014). The model treats Quizbowl as classification and predicts the answer entity from a sequence of words represented as 300-dimensional GloVe embeddings (Pennington et al., 2014). Both models in this round are trained using an expanded dataset of approximately 110,000 Quizbowl questions. We expanded the round two dataset to incorporate more diverse answers (25,000 entities versus 11,000 in round one).

3.3 Interpreting Quizbowl Models

To help write adversarial questions, we expose what the model is thinking to the authors. We interpret models using saliency heat maps: each word of the question is highlighted based on its importance to the model's prediction (Ribeiro et al., 2016).

For the neural model, word importance is the decrease in prediction probability when a word is removed (Li et al., 2016; Wallace et al., 2018). We focus on gradient-based approximations (Simonyan et al., 2014; Montavon et al., 2017) for their computational efficiency.

To interpret a model prediction on an input sequence of n words $\mathbf{w} = \langle \boldsymbol{w}_1, \boldsymbol{w}_2, \ldots \boldsymbol{w}_n \rangle$, we approximate the classifier f with a linear function of w_i derived from the first-order Taylor expansion. The importance of w_i, with embedding $\boldsymbol{v}_i$, is the derivative of f with respect to the one-hot vector:

$$\frac{\partial f}{\partial w_i} = \frac{\partial f}{\partial \boldsymbol{v}_i} \frac{\partial \boldsymbol{v}_i}{\partial w_i} = \frac{\partial f}{\partial \boldsymbol{v}_i} \cdot \boldsymbol{v}_i. \quad (1)$$

This simulates how model predictions change when a particular word's embedding is set to the zero vector, i.e., it approximates word removal (Ebrahimi et al., 2018; Wallace et al., 2018).

For the IR model, we use the ElasticSearch Highlight API (Gormley and Tong, 2015), which provides word importance scores based on query matches from the inverted index.

3.4 Adversarial Writing Interface

The authors interact with either the IR or RNN model through a user interface[1] (Figure 3). An author writes their question in the upper right while the model's top five predictions (*Machine Guesses*) appear in the upper left. If the top prediction is

---

[1] https://github.com/Eric-Wallace/trickme-interface/

Figure 3: The author writes a question (top right), the QA system provides guesses (left), and explains why it makes those guesses (bottom right). The author can then adapt their question to "trick" the model.

the right answer, the interface indicates where in the question the model is first correct. The goal is to cause the model to be incorrect or to delay the correct answer position as much as possible.[2] The words of the current question are highlighted using the applicable interpretation method in the lower right (*Evidence*). We do not enforce time restrictions or require questions to be adversarial: if the author fails to break the system, they are free to "give up" and submit any question.

The interface continually updates as the author writes. We track the question edit history to identify recurring model failures (Section 6) and understand how interpretations guide the authors (Section 7).

3.5 Question Authors

We focus on members of the Quizbowl community: they have deep trivia knowledge and craft questions for Quizbowl tournaments (Jennings, 2006). We award prizes for questions read at live human–computer matches (Section 5.3).

The question authors are familiar with the standard format of Quizbowl questions (Lujan and Teitler, 2003). The questions follow a common paragraph structure, are well edited for grammar,

and finish with a simple "give-away" clue. These constraints benefit the adversarial writing process as it is very clear what constitutes a difficult but valid question. Thus, our examples go beyond surface level "breaks" such as character noise (Belinkov and Bisk, 2018) or syntax changes (Iyyer et al., 2018). Rather, questions are difficult because of their semantic content (examples in Section 6).

3.6 How an Author Writes a Question

To see how an author might write a question with the interface, we walk through an example of writing a question's first sentence. The author first selects the answer to their question from the training set—Johannes Brahms—and begins:

> Karl Ferdinand Pohl showed this composer some pieces on which this composer's Variations on a Theme by Haydn were based.

The QA system *buzzes* (i.e., it has enough information to interrupt and answer correctly) after "composer". The author sees that the name "Karl Ferdinand Pohl" appears in Brahms' Wikipedia page and avoids that specific phrase, describing Pohl's position instead of naming him directly:

> This composer was given a theme called "Chorale St. Antoni" by the archivist of the Vienna Musikverein, which could have been written by Ignaz Pleyel.

---

[2]The authors want normal Quizbowl questions which humans can easily answer by the very end. For popular answers, (e.g., Australia or Suez Canal), writing novel final give-away clues is difficult. We thus expect models to often answer correctly by the very end of the question.

Science	17%
History	22%
Literature	18%
Fine Arts	15%
Religion, Mythology, Philosophy, and Social Science	13%
Current Events, Geography, and General Knowledge	15%
Total Questions	1213

Table 1: The topical diversity of the questions in the adversarially-authored dataset based on a random sample of 100 questions.

This rewrite adds in some additional information (there is a scholarly disagreement over who wrote the theme and its name), and the QA system now incorrectly thinks the answer is Frédéric Chopin. The user can continue to build on the theme, writing

> While summering in Tutzing, this composer turned that theme into "Variations on a Theme by Haydn".

Again, the author then sees that the system buzzes "Variations on a Theme" with the correct answer. However, the author can rewrite it in its original German, "Variationen über ein Thema von Haydn" to fool the system. The author continues to create entire questions the model cannot solve.

4 A New Adversarially-Authored Dataset

Our adversarial dataset consists of 1213 questions with 6,541 sentences across diverse topics (Table 1).[3] There are 807 questions written against the IR system and 406 against the neural model by 115 unique authors. We plan to hold twice-yearly competitions to continue data collection.

4.1 Validating Questions with Quizbowlers

We validate that the adversarially-authored questions are not of poor quality or too difficult for humans. We first automatically filter out questions based on length, the presence of vulgar statements, or repeated submissions (including re-submissions from the Quizbowl training or evaluation data).

We next host a human-only Quizbowl event using intermediate and expert players (former and current collegiate Quizbowl players). We select sixty adversarially-authored questions and sixty

---

[3]Data available at `http://trickme.qanta.org`.

standard high school national championship questions, both with the same number of questions per category (list of categories in Table 1).

To answer a Quizbowl question, a player interrupts the question: the earlier the better. To capture this dynamic, we record both the average answer position (as a percentage of the question, lower is better) and answer accuracy. We shuffle the regular and adversarially-authored questions, read them to players, and record these two metrics.

The adversarially-authored questions are on average *easier* for humans than the regular test questions. For the adversarially-authored set, humans buzz with 41.6% of the question remaining and an accuracy of 89.7%. On the standard questions, humans buzz with 28.3% of the question remaining and an accuracy of 84.2%. The difference in accuracy between the two types of questions is not significantly different ($p = 0.16$ using Fisher's exact test), but the buzzing position is earlier for adversarially-authored questions ($p = 0.0047$ for a two-sided t-test). We expect the questions that were not played to be of comparable difficulty because they went through the same submission process and post-processing. We further explore the human-perceived difficulty of the adversarially-authored questions in Section 5.3.

5 Computer Experiments

This section evaluates QA systems on the adversarially-authored questions. We test three models: the IR and RNN models shown in the interface, as well as a Deep Averaging Network (Iyyer et al., 2015, DAN) to evaluate the transferability of the adversarial questions. We break our study into two rounds. The first round consists of adversarially-authored questions written against the IR system (Section 5.1); the second round questions target both the IR and RNN (Section 5.2).

Finally, we also hold live competitions that pit the state-of-the-art Studio Ousia model (Yamada et al., 2018) against human teams (Section 5.3).

5.1 First Round Attacks: IR Adversarial Questions Transfer To All Models

The first round of adversarially-authored questions target the IR model and are significantly harder for the IR, RNN, and DAN models (Figure 4). For example, the DAN's accuracy drops from 54.1% to 32.4% on the full question (60% of original performance).

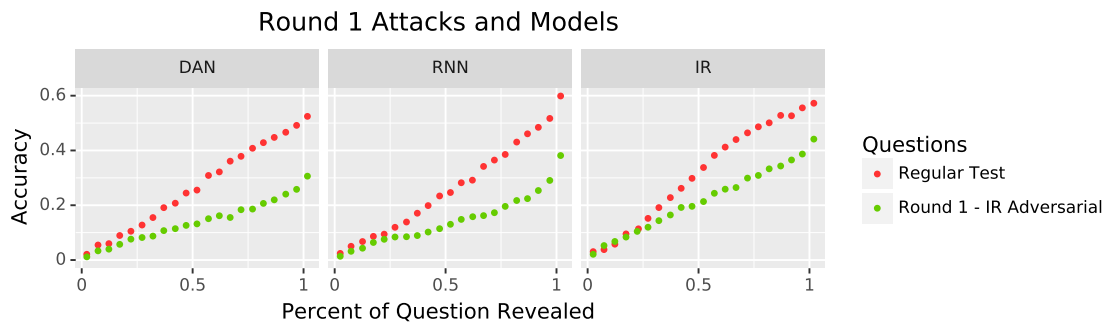For both adversarially-authored and original test

Figure 4: The first round of adversarial writing attacks the IR model. Like regular test questions, adversarially-authored questions begin with difficult clues that trick the model. However, the adversarial questions are significantly harder during the crucial middle third of the question.
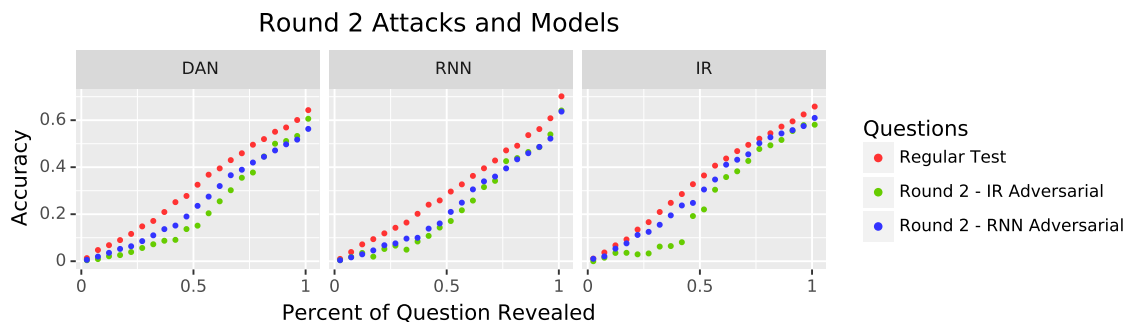

Figure 5: The second round of adversarial writing attacks the IR and RNN models. The questions targeted against the IR system degrade the performance of all models. However, the reverse does not hold: the IR model is robust to the questions written to fool the RNN.

questions, the early clues are difficult to answer (accuracy about 10% through 25% of the question). However, during the middle third of the questions, where buzzes in Quizbowl most frequently occur, the accuracy on original test questions rises significantly quicker than the adversarially-authored ones. For both type of questions, the accuracy rises towards the end as the clues become "give-aways".

5.2 Second Round Attacks: RNN Adversarial Questions are Brittle

In the second round, the authors also attack an RNN model. All models tested in the second round are trained on a larger dataset (Section 3.2).

A similar trend holds for IR adversarial questions in the second round (Figure 5): a question that tricks the IR system also fools the two neural models (i.e., adversarial examples transfer). For example, the DAN model was never targeted but had substantial accuracy decreases in both rounds.

However, this does not hold for questions written adversarially against the RNN model. On these questions, the neural models struggle but the IR model is largely unaffected (Figure 5, right).

5.3 Humans vs. Computer, Live!

In the offline setting (i.e., no pressure to "buzz" before an opponent) models demonstrably struggle on the adversarial questions. But, what happens in standard Quizbowl: live, head-to-head games?

We run two live humans vs. computer matches. The first match uses IR adversarial questions in a forty question, tossup-only Quizbowl format. We pit a human team of national-level Quizbowl players against the Studio Ousia model (Yamada et al., 2018), the current state-of-the-art Quizbowl system. The model combines neural, IR, and knowledge graph components (details in Appendix B), and won the 2017 NIPS shared task, defeating a team of expert humans 475–200 on regular Quizbowl test questions. Although the team at our live event was comparable to the NIPS 2017 team, the tables were turned: the human team won handedly 300–30.

Our second live event is significantly larger: seven human teams play against models on over 400 questions written adversarially against the RNN model. The human teams range in ability from high school Quizbowl players to national-level teams
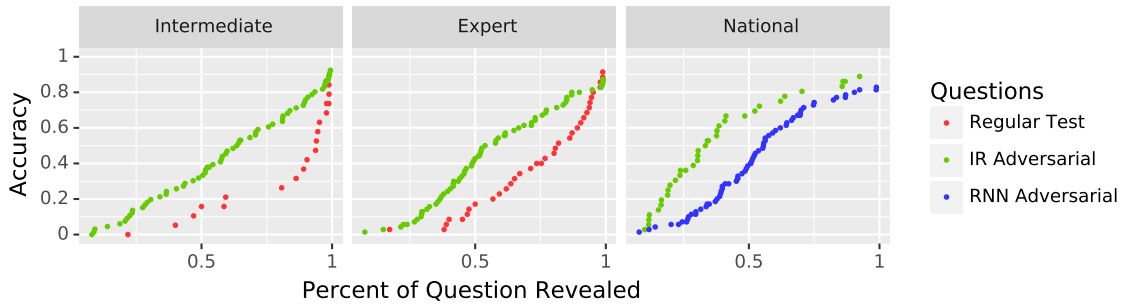
Figure 6: Humans find adversarially-authored question about as difficult as normal questions: rusty weekend warriors (*Intermediate*), active players (*Expert*), or the best trivia players in the world (*National*).
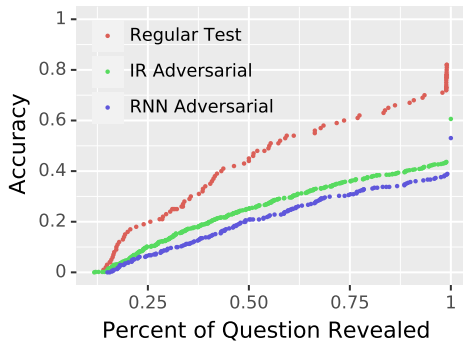

Figure 7: The accuracy of the state-of-the-art Studio Ousia model degrades on the adversarially-authored questions despite never being directly targeted. This verifies that our findings generalize beyond the RNN and IR models.

(Jeopardy! champions, Academic Competition Federation national champions, top scorers in the World Quizzing Championships). The models are based on either IR or neural methods. Despite a few close games between the weaker human teams and the models, humanity prevailed in every match.[4]

Figures 6–7 summarize the live match results for the humans and Ousia model, respectively. Humans and models have considerably different trends in answer accuracy. Human accuracy on both regular and adversarial questions rises quickly in the *last half* of the question (curves in Figure 6). In essence, the "give-away" clues at the end of questions are easy for humans to answer.

On the other hand, models on regular test questions do well in the *first half*, i.e., the "difficult" clues for humans are easier for models (*Regular Test* in Figure 7). However, models, like humans, struggle on adversarial questions in the first half.

---

6 What Makes Adversarially-authored Questions Hard?

This section analyzes the adversarially-authored questions to identify the source of their difficulty.

6.1 Quantitative Differences in Questions

One possible source of difficulty is data scarcity: the answers to adversarial questions rarely appear in the training set. However, this is not the case; the mean number of training examples per answer (e.g., George Washington) is 14.9 for the adversarial questions versus 16.9 for the regular test data.

Another explanation for question difficulty is limited "overlap" with the training data, i.e., models cannot match n-grams from the training clues. We measure the proportion of test n-grams that also appear in training questions with the same answer (Table 2). The overlap is roughly equal for unigrams but surprisingly higher for adversarial questions' bigrams. The adversarial questions are also shorter and have fewer NEs. However, the proportion of named entities is roughly equivalent.

One difference between the questions written against the IR system and the ones written against the RNN model is the drop in NEs. The decrease in NEs is higher for IR adversarial questions, which may explain their generalization: the RNN is more sensitive to changes in phrasing, while the IR system is more sensitive to specific words.

6.2 Categorizing Adversarial Phenomena

We next qualitatively analyze adversarially-authored questions. We manually inspect the author edit logs, classifying questions into six different phenomena in two broad categories (Table 3) from a random sample of 100 questions, double counting questions into multiple phenomena when applicable.

	Adversarial	Regular
Unigram overlap	0.40	0.37
Bigram overlap	0.08	0.05
Longest n-gram overlap	6.73	6.87
Average NE overlap	0.38	0.46
IR Adversarial	0.35	
RNN Adversarial	0.44	
Total Words	107.1	133.5
Total NE	9.1	12.5

Table 2: The adversarially-authored questions have similar n-gram overlap to the regular test questions. However, the overlap of the named entities (NE) decreases for IR Adversarial questions.

Composing Seen Clues	15%
Logic & Calculations	5%
Multi-Step Reasoning	25%
Paraphrases	38%
Entity Type Distractors	7%
Novel Clues	26%
Total Questions	1213

Table 3: A breakdown of the phenomena in the adversarially-authored dataset.

6.2.1 Adversarial Category 1: Reasoning

The first question category requires reasoning about known clues (Table 4).

Composing Seen Clues: These questions provide entities with a first-order relationship to the correct answer. The system must triangulate the correct answer by "filling in the blank". For example, the first question of Table 4 names the place of death of Tecumseh. The training data contains a question about his death reading "though stiff fighting came from their Native American allies under Tecumseh, who died at this battle" (The Battle of the Thames). The system must connect these two clues to answer.

Logic & Calculations: These questions require mathematical or logical operators. For example, the training data contains a clue about the Battle of Thermopylae: "King Leonidas and 300 Spartans died at the hands of the Persians". The second question in Table 4 requires adding 150 to the number of Spartans.

Multi-Step Reasoning: This question type requires multiple reasoning steps between entities. For example, the last question of Table 4 requires a reasoning step from the "I Have A Dream" speech to the Lincoln Memorial and then another reasoning step to reach Abraham Lincoln.

6.2.2 Adversarial Category 2: Distracting Clues

The second category consists of circumlocutory clues (Table 5).

Paraphrases: A common adversarial modification is to paraphrase clues to remove exact n-gram matches from the training data. This renders our IR system useless but also hurts the neural models. Many of the adversarial paraphrases go beyond syntax-only changes (e.g., the first row of Table 5).

Entity Type Distractors: Whether explicit or implicit in a model, one key component for QA is determining the answer type of the question. Authors take advantage of this by providing clues that cause the model to select the wrong answer type. For example, in the second question of Table 5, the "lead-in" clue implies the answer may be an actor. The RNN model answers Don Cheadle in response despite previously seeing the Bill Clinton "playing a saxophone" clue in the training data.

Novel Clues: Some adversarially-authored questions are hard not because of phrasing or logic but because our models have not seen these clues. These questions are easy to create: users can add *Novel Clues* that—because they are not uniquely associated with an answer—confuse the models. While not as linguistically interesting, novel clues are not captured by Wikipedia or Quizbowl data, thus improving the dataset's diversity. For example, adding clues about literary criticism (Hardwick, 1967; Watson, 1996) to a question about Lillian Hellman's The Little Foxes: "Ritchie Watson commended this play's historical accuracy for getting the price for a dozen eggs right—ten cents—to defend against Elizabeth Hardwick's contention that it was a sentimental history." Novel clues create an incentive for models to use information beyond past questions and Wikipedia.

Novel clues have different effects on IR and neural models: while IR models largely ignore them, novel clues can lead neural models astray. For example, on a question about Tiananmen Square, the RNN model buzzes on the clue "World Economic

Question	Prediction	Answer	Phenomenon
This man, who died at the Battle of the Thames, experienced a setback when his brother Tenskwatawa's influence over their tribe began to fade.	Battle of Tippecanoe	Tecumseh	Composing Seen Clues
This number is one hundred fifty more than the number of Spartans at Thermopylae.	Battle of Thermopylae	450	Logic & Calculations
A building dedicated to this man was the site of the "I Have A Dream" speech.	Martin Luther King Jr.	Abraham Lincoln	Multi-Step Reasoning

Table 4: The first category of adversarially-authored questions consists of examples that require reasoning. *Answer* displays the correct answer (all models were incorrect). For these examples, connecting the training and adversarially-authored clues is simple for humans but difficult for models.

Set	Question	Prediction	Phenomenon
Training	Name this sociological phenomenon, the *taking of one's own life*.	Suicide	Paraphrase
Adversarial	Name this *self-inflicted method of death*.	Arthur Miller	
Training	Clinton played the *saxophone on The Arsenio Hall Show*.	Bill Clinton	
Adversarial	He was edited to appear in the film "Contact"… For ten points, name this American president who played the *saxophone on an appearance on the Arsenio Hall Show*.	Don Cheadle	Entity Type Distractor

Table 5: The second category of adversarial questions consists of clues that are present in the training data but are written in a distracting manner. *Training* shows relevant snippets from the training data. *Prediction* displays the RNN model's answer prediction (always correct on Training, always incorrect on Adversarial).

Herald". However, adding a novel clue about "the history of shaving" renders the brittle RNN unable to buzz on the "World Economic Herald" clue that it was able to recognize before.[5] This helps to explain why adversarially-authored questions written against the RNN do not stump IR models.

7   How Do Interpretations Help?

This section explores how model interpretations help to guide adversarial authors. We analyze the question edit log, which reflects how authors modify questions given a model interpretation.

A direct edit of the highlighted words often creates an adversarial example (e.g., Figure 8). Figure 9 shows a more intricate example. The left plot shows the *Question Length*, as well as the position where the model is first correct (*Buzzing Position*, lower is better). We show two adversarial edits. In the first (**1**), the author removes the first sentence of the question, which makes the question *easier* for

> One of these concepts … a **Hyperbola** is a type of, for ten points, what shapes made by passing a **plane** through a namesake solid, ~~that also includes the **ellipse**, **parabola**?~~ whose area is given by one-third Pi r squared times height?
> *Prediction*: Conic Section (✓) → Sphere (✗)

Figure 8: The interpretation successfully aids an attack against the IR system. The author removes the phrase containing the words "ellipse" and "parabola", which are highlighted in the interface (shown in bold). In its place, they add a phrase which the model associates with the answer sphere.

the model (buzz position decreases). The author counteracts this in the second edit (**2**), where they use the interpretation to craft a targeted modification which breaks the IR model.

However, models are not always this brittle. In Figure 10 (Appendix C), the interpretation fails to aid an adversarial attack against the RNN model. At each step, the author uses the highlighted words as a guide to edit targeted portions of the question yet

---

[5]The "history of shaving" is a tongue-in-cheek name for a poster displaying the hirsute leaders of Communist thought. It goes from the bearded Marx and Engels, to the mustachioed Lenin and Stalin, and finally the clean-shaven Mao.

Figure 9: The *Question Length* and the position where the model is first correct (*Buzzing Position*, lower is better) are shown as a question is written. In (**1**), the author makes a mistake by removing a sentence that makes the question easier for the IR model. In (**2**), the author uses the interpretation, replacing the highlighted word (shown in bold) "molecules" with "species" to trick the RNN model.

fails to trick the model. The author gives up and submits their relatively non-adversarial question.

7.1 Interviews With Adversarial Authors

We also interview the adversarial authors who attended our live events. Multiple authors agree that identifying oft-repeated "stock" clues was the interface's most useful feature. As one author explained, "There were clues which I did not think were stock clues but were later revealed to be". In particular, the author's question about the Congress of Vienna used a clue about "Kraków becoming a free city", which the model immediately recognized.

Another interviewee was Jordan Brownstein,[6] a national Quizbowl champion and one of the best active players, who felt that computer opponents were better at questions that contained direct references to battles or poetry. He also explained how the different writing styles used by each Quizbowl author increases the difficulty of questions for computers. The interface's evidence panel allows authors to read existing clues which encourages these unique stylistic choices.

8 Related Work

New datasets often allow for a finer-grained analysis of a linguistic phenomenon, task, or genre. The LAMBADA dataset (Paperno et al., 2016) tests a model's understanding of the broad contexts

present in book passages, while the Natural Questions corpus (Kwiatkowski et al., 2019) combs Wikipedia for answers to questions that users trust search engines to answer (Oeldorf-Hirsch et al., 2014). Other work focuses on natural language inference, where challenge examples highlight model failures (Wang et al., 2019; Glockner et al., 2018; Naik et al., 2018). Our work is unique in that we use human adversaries to expose model weaknesses, which provides a diverse set of phenomena (from paraphrases to multi-hop reasoning) that models cannot solve.

Other work puts an adversary in the data annotation or postprocessing loop. For instance, Dua et al. (2019) and Zhang et al. (2018) filter out easy questions using a baseline QA model, while Zellers et al. (2018) use stylistic classifiers to filter language inference examples. Rather than filtering out easy questions, we instead use human adversaries to generate hard ones. Similar to our work, Ettinger et al. (2017) use human adversaries. We extend their setting by providing humans with model interpretations to facilitate adversarial writing. Moreover, we have a ready-made audience of question writers to generate adversarial questions.

The collaborative adversarial writing process reflects the complementary abilities of humans and computers. For instance, "centaur" chess teams of both a human and a computer are often stronger than a human or computer alone (Case, 2018). In Starcraft, humans devise high-level "macro" strategies, while computers are superior at executing fast

---

[6] https://www.qbwiki.com/wiki/Jordan_Brownstein

and precise "micro" actions (Vinyals et al., 2017). In NLP, computers aid simultaneous human interpreters (He et al., 2016) at remembering forgotten information or translating unfamiliar words.

Finally, recent approaches to adversarial evaluation of NLP models (Section 2) typically target one phenomenon (e.g., syntactic modifications) and complement our human-in-the-loop approach.

9 Conclusion

One of the challenges of machine learning is knowing why systems fail. This work brings together two threads that attempt to answer this question: visualizations and adversarial examples. Visualizations underscore the capabilities of existing models, while adversarial examples—crafted with the ingenuity of human experts—show that these models are still far from matching human prowess.

Our experiments with both neural and IR methodologies show that QA models still struggle with synthesizing clues, handling distracting information, and adapting to unfamiliar data. Our adversarially-authored dataset is only the first of many iterations (Ruef et al., 2016): as models improve, future adversarially-authored datasets can elucidate the limitations of next-generation QA systems.

While we focus on QA, our procedure is applicable to other NLP settings where there is (1) a pool of talented authors who (2) write text with specific goals. Future research can look to craft adversarially-authored datasets for other NLP tasks that meet these criteria.

References

Yonatan Belinkov and Yonatan Bisk. 2018. Synthetic and natural noise both break neural machine translation. In *Proceedings of the International Conference on Learning Representations*.

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

Jordan Boyd-Graber, Shi Feng, and Pedro Rodriguez. 2018. *Human-Computer Question Answering: The Case for Quizbowl*. Springer.

Nicky Case. 2018. How To Become A Centaur. *Journal of Design and Science*.

Danqi Chen, Jason Bolton, and Christopher D. Manning. 2016. A thorough examination of the CNN/Daily Mail reading comprehension task. *Proceedings of the Association for Computational Linguistics*.

Kyunghyun Cho, Bart van Merrienboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning phrase representations using RNN encoder-decoder for statistical machine translation. In *Proceedings of Empirical Methods in Natural Language Processing*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Conference of the North American Chapter of the Association for Computational Linguistics*.

Dheeru Dua, Yizhong Wang, Pradeep Dasigi, Gabriel Stanovsky, Sameer Singh, and Matt Gardner. 2019. DROP: A reading comprehension benchmark requiring discrete reasoning over paragraphs. In *Conference of the North American Chapter of the Association for Computational Linguistics*.

Javid Ebrahimi, Anyi Rao, Daniel Lowd, and Dejing Dou. 2018. HotFlip: White-box adversarial examples for text classification. In *Proceedings of the Association for Computational Linguistics*.

Allyson Ettinger, Sudha Rao, Hal Daumé III, and Emily M. Bender. 2017. Towards linguistically generalizable NLP systems: A workshop and shared task. In *In Proceedings of the First Workshop on Building Linguistically Generalizable NLP Systems*.

David Ferrucci, Eric Brown, Jennifer Chu-Carroll, James Fan, David Gondek, Aditya A. Kalyanpur, Adam Lally, J. William Murdock, Eric Nyberg, John Prager, Nico Schlaefer, and Chris Welty. 2010. Building Watson: An Overview of the DeepQA Project. *AI Magazine*, 31(3).

Max Glockner, Vered Shwartz, and Yoav Goldberg. 2018. Breaking NLI systems with sentences that require simple lexical inferences. In *Proceedings of the Association for Computational Linguistics*.

Clinton Gormley and Zachary Tong. 2015. *Elasticsearch: The Definitive Guide*. O'Reilly Media, Inc.

Anupam Guha, Mohit Iyyer, Danny Bouman, and Jordan Boyd-Graber. 2015. Removing the training wheels: A coreference dataset that entertains humans and challenges computers. In *North American Association for Computational Linguistics*.

Suchin Gururangan, Swabha Swayamdipta, Omer Levy, Roy Schwartz, Samuel R. Bowman, and Noah A. Smith. 2018. Annotation artifacts in natural language inference data. In *Conference of the North American Chapter of the Association for Computational Linguistics*.

Elizabeth Hardwick. 1967. The Little Foxes revived. *The New York Review of Books*, 9(11).

Bob Harris. 2006. *Prisoner of Trebekistan: A Decade in Jeopardy!*

He He, Jordan Boyd-Graber, and Hal Daumé III. 2016. Interpretese vs. translationese: The uniqueness of human strategies in simultaneous interpretation. In *Conference of the North American Chapter of the Association for Computational Linguistics*.

Lynette Hirschman and Rob Gaizauskas. 2001. Natural language question answering: The view from here. *Natural Language Engineering*, 7(4):275–300.

Mohit Iyyer, Varun Manjunatha, Jordan Boyd-Graber, and Hal Daumé III. 2015. Deep unordered composition rivals syntactic methods for text classification. In *Proceedings of the Association for Computational Linguistics*.

Mohit Iyyer, John Wieting, Kevin Gimpel, and Luke Zettlemoyer. 2018. Adversarial example generation with syntactically controlled paraphrase networks. In *Conference of the North American Chapter of the Association for Computational Linguistics*.

Ken Jennings. 2006. *Brainiac: adventures in the curious, competitive, compulsive world of trivia buffs*. Villard.

Robin Jia and Percy Liang. 2017. Adversarial examples for evaluating reading comprehension systems. In *Proceedings of Empirical Methods in Natural Language Processing*.

Ike Jose. 2017. The craft of writing pyramidal quiz questions: Why writing quiz bowl questions is an intellectual task.

Divyansh Kaushik and Zachary C. Lipton. 2018. How much reading does reading comprehension require? A critical investigation of popular benchmarks. In *Proceedings of Empirical Methods in Natural Language Processing*.

Tom Kwiatkowski, Jennimaria Palomaki, Olivia Rhinehart, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Matthew Kelcey, Jacob Devlin, et al. 2019. Natural Questions: a benchmark for question answering research. In *Transactions of the Association for Computational Linguistics*.

Jiwei Li, Will Monroe, and Dan Jurafsky. 2016. Understanding neural networks through representation erasure. *arXiv preprint arXiv:1612.08220*.

Paul Lujan and Seth Teitler. 2003. Writing good quizbowl questions: A quick primer.

Grégoire Montavon, Wojciech Samek, and Klaus-Robert Müller. 2017. Methods for interpreting and understanding deep neural networks. *arXiv preprint*, abs/1706.07979.

Pramod Kaushik Mudrakarta, Ankur Taly, Mukund Sundararajan, and Kedar Dhamdhere. 2018. Did

the model understand the question? In *Proceedings of the Association for Computational Linguistics*.

Aakanksha Naik, Abhilasha Ravichander, Norman Sadeh, Carolyn Rose, and Graham Neubig. 2018. Stress test evaluation for natural language inference. In *Proceedings of International Conference on Computational Linguistics*.

Anne Oeldorf-Hirsch, Brent Hecht, Meredith Ringel Morris, Jaime Teevan, and Darren Gergle. 2014. To search or to ask: the routing of information needs between traditional search engines and social networks. In *Conference on Computer Supported Cooperative Work and Social Computing*.

Denis Paperno, Germán Kruszewski, Angeliki Lazaridou, Quan Ngoc Pham, Raffaella Bernardi, Sandro Pezzelle, Marco Baroni, Gemma Boleda, and Raquel Fernández. 2016. The LAMBADA dataset: Word prediction requiring a broad discourse context. In *Proceedings of the Association for Computational Linguistics*.

Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. GloVe: Global vectors for word representation. In *Proceedings of Empirical Methods in Natural Language Processing*.

Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. 2016. Why should I trust you?: Explaining the predictions of any classifier. In *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*.

Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. 2018. Semantically equivalent adversarial rules for debugging nlp models. In *Proceedings of the Association for Computational Linguistics*.

Andrew Ruef, Michael Hicks, James Parker, Dave Levin, Michelle L. Mazurek, and Piotr Mardziel. 2016. Build it, break it, fix it: Contesting secure development. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*.

Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. 2014. Deep inside convolutional networks: Visualising image classification models and saliency maps. In *Proceedings of the International Conference on Learning Representations*.

Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian J. Goodfellow, and Rob Fergus. 2013. Intriguing properties of neural networks. In *Proceedings of the International Conference on Learning Representations*.

Oriol Vinyals, Timo Ewalds, Sergey Bartunov, Petko Georgiev, Alexander Sasha Vezhnevets, Michelle Yeo, Alireza Makhzani, Heinrich Küttler, John Agapiou, Julian Schrittwieser, John Quan, Stephen Gaffney, Stig Petersen, Karen Simonyan, Tom Schaul, Hado van Hasselt, David Silver, Timothy P. Lillicrap, Kevin Calderone, Paul Keet, Anthony Brunasso, David Lawrence, Anders Ekermo, Jacob Repp, and Rodney Tsing. 2017. Starcraft II: A new challenge for reinforcement learning. *arXiv preprint arXiv:1708.04782*.

Eric Wallace, Shi Feng, and Jordan Boyd-Graber. 2018. Interpreting neural networks with nearest neighbors. In *EMNLP 2018 Workshop on Analyzing and Interpreting Neural Networks for NLP*.

Alex Wang, Amapreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R. Bowman. 2019. Glue: A multi-task benchmark and analysis platform for natural language understanding. In *Proceedings of the International Conference on Learning Representations*.

Ritchie D. Watson. 1996. Lillian Hellman's "The Little Foxes" and the new south creed: An ironic view of southern history. *The Southern Literary Journal*, 28(2):59–68.

Ikuya Yamada, Ryuji Tamaki, Hiroyuki Shindo, and Yoshiyasu Takefuji. 2018. Studio ousia's quiz bowl question answering system. *arXiv preprint arXiv:1803.08652*.

Adams Wei Yu, David Dohan, Minh-Thang Luong, Rui Zhao, Kai Chen, Mohammad Norouzi, and Quoc V. Le. 2018. QANet: Combining local convolution with global self-attention for

reading comprehension. In *Proceedings of the International Conference on Learning Representations*.

Rowan Zellers, Yonatan Bisk, Roy Schwartz, and Yejin Choi. 2018. SWAG: A large-scale adversarial dataset for grounded commonsense inference. In *Proceedings of Empirical Methods in Natural Language Processing*.

Sheng Zhang, Xiaodong Liu, Jingjing Liu, Jianfeng Gao, Kevin Duh, and Benjamin Van Durme. 2018. Record: Bridging the gap between human and machine commonsense reading comprehension. *arXiv preprint arXiv:1810.12885*.

Zhengli Zhao, Dheeru Dua, and Sameer Singh. 2018. Generating natural adversarial examples. In *Proceedings of the International Conference on Learning Representations*.

Sentence	Success/Failure Phenomena
its types include "frictional", "cyclical", and "structural" its types include "frictional", and structural	Missing Information ✗
german author of the sorrows of young werther and a two-part faust german author of the sorrows of mr. werther	Lost Named Entity ✗
name this elegy on the death of john keats composed by percy shelley name was this elegy on the death of percy shelley	Incorrect Clue ✗
identify this play about willy loman written by arthur miller so you can identify this work of mr. miller	Unsuited Syntax Template ✗
he employed marco polo and his father as ambassadors he hired marco polo and his father as ambassadors	Verb Synonym ✓

Table 6: Failure and success cases for SCPN. The model fails to create a valid paraphrase of the sentence for 97% of questions.

A Failure of Syntactically Controlled Paraphrase Networks

We apply the Syntactically Controlled Paraphrase Network (Iyyer et al., 2018, SCPN) to Quizbowl questions. The model operates on the sentence level and cannot paraphrase paragraphs. We thus feed in each sentence independently, ignoring possible breaks in coreference. The model does not correctly paraphrase most of the complex sentences present in Quizbowl questions. The paraphrases were rife with issues: ungrammatical, repetitive, or missing information.

To simplify the setting, we focus on paraphrasing the shortest sentence from each question (often the final clue). The model still fails in this case. We analyze a random sample of 200 paraphrases: only six maintained all of the original information.

Table 6 shows common failure cases. One recurring issue is an inability to maintain the correct named entities after paraphrasing. In Quizbowl, maintaining entity information is vital for ensuring question validity. We were surprised by this failure because SCPN incorporates a copy mechanism.

B Studio Ousia Quizbowl Model

The Studio Ousia system works by aggregating scores from both a neural text classification model and an IR system. Additionally, it scores answers based on their match with the correct entity type (e.g., religious leader, government agency, etc.) predicted by a neural entity type classifier. The Studio Ousia system also uses data beyond Quizbowl questions and the text of Wikipedia pages, integrating entities from a knowledge graph and customized word vectors (Yamada et al., 2018).

C Failed Adversarial Attempt

Figure 10 shows a user's failed attempt to break the neural Quizbowl model.

In his speeches this . . . As a Senator, this man supported **Paraguay** in the **Chaco War**, believing **Bolivia** was backed by Standard Oil. this man's campaign was endorsed by Milo Reno and Charles Coughlin.
Prediction: Huey Long (✓) → Huey Long (✓)

In his speeches this . . . As a Senator, this man's campaign was endorsed by **Milo Reno** and **Charles Coughlin**. a Catholic priest and radio show host.
Prediction: Huey Long (✓) → Huey Long (✓)

Figure 10: A failed attempt to trick the neural model. The author modifies the question multiple times, replacing words suggested by the interpretation, but is unable to break the system.