Trick Me If You Can: Adversarial Writing of Trivia Challenge Questions

Eric Wallace, Pedro Rodriguez, Shi Feng, Jordan Boyd-Graber
University of Maryland
{ewallac2,entilzha,shifeng,jbg}@umiacs.umd.edu

Abstract

Modern natural language processing systems have been touted as approaching human performance. However, existing datasets are imperfect tests. Examples are written with humans in mind, not computers, and often do not properly expose model limitations. We address this by developing a new process for crowdsourced annotation, adversarial writing, where humans interact with trained models and try to break them. Applying this annotation process to Trivia question answering yields a challenge set, which despite being easy for human players to answer, systematically stumps automated question answering systems. Diagnosing model errors on the evaluation data provides actionable insights to explore in developing more robust and generalizable question answering systems.

1 Introduction

Proponents of modern machine learning systems have claimed human parity on difficult tasks such as question answering (Yu et al., 2018). Datasets such as SQuAD and TriviaQA (Rajpurkar et al., 2016; Joshi et al., 2017) certainly advanced the state of the art, but do they provide the right examples to measure how well machines can answer questions?

Many existing language datasets are written and evaluated with humans in mind, not computers. However, computers solve NLP tasks in a fundamentally different way than humans, training on thousands of examples, rather than analyzing examples in isolation. This allows models to pick up on superficial patterns that may occur in data crawled from the internet (Chen et al., 2016) or from biases in the crowdsourced annotation process (Gururangan et al., 2018). Additionally, because existing test sets do not provide specific diagnostic information, it is difficult to get the proper insight into a system's capabilities and limitations. Unfortunately, when rigorous evaluations are not performed, strikingly simple model limitations can be overlooked (Belinkov and Bisk, 2018; Jia and Liang, 2017).

To address this, we create a new framework for crowdsourced annotation that provides writers with model interpretations to adversarially craft examples. These interpretations are presented in an interactive user interface (Section 2) to facilitate a model-driven data annotation process. We apply this specifically to Question Answering (QA), where we ask trivia enthusiasts—who write new questions for scholastic and open circuit tournaments—to create examples that challenge existing QA models.

Humans find the resulting challenge questions *easier* than regular questions (Section 3), but the accuracy of strong QA models decreases as much as 40%. (Section 4). In a live match using the challenge questions, a state of the art QA system scores nearly zero points against a human team, despite defeating comparable teams in past events (Section 4.1). Unlike many existing QA test sets, our questions highlight specific phenomena that humans can capture but machines cannot (Section 5). We release code for the annotation interface and our challenge set to better evaluate and systematically improve models.[1]

[1] www.qanta.org

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 Giacomo Puccini opera about an American lieutenants affair with the Japanese woman Cio-Cio San.

ANSWER: Madama Butterfly

Figure 1: An example Quizbowl question. The question becomes progressively easier to answer later on; thus, more knowledgeable players can answer after hearing fewer clues.

2 A Model-Driven Annotation Process

This section introduces our framework for tailoring questions to challenge computers, the surrounding community of trivia enthusiasts that create thousands of questions annually, and how we expose QA algorithms to this community to help them craft questions that challenge computers.

2.1 The Quizbowl Trivia Task

The "gold standard" of academic competitions between universities and high schools is Quizbowl. Unlike other QA formats such as Jeopardy! or TriviaQA (Joshi et al., 2017), 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": questions start with difficult clues and get progressively easier. These "pyramidal" style questions (Jose, 2017) are carefully crafted by the authors to allow the most knowledgeable player to answer first. A question on Paris that begins "this capital of France" would be a test of reaction speed, not knowledge; thus, skilled question writers arrange the clues such that players will recognize them with increasing probability (Figure 1).

However, like most existing QA datasets, Quizbowl questions are written with humans in mind. Unfortunately, the heuristics that question writers 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 baseline QA system easily solves the example in Figure 1 from seeing the reference to "Un Bel Di". Other questions contain uniquely identifying "trigger words". For example, "martensite" only appears in questions on steel. For these types of examples, a QA system needs to understand no additional information other than an if–then rule. Surprisingly, this is true for many different answers: in these cases, QA devolves into trivial pattern matching. Consequently, information retrieval systems are strong baselines for this task, even capable of defeating top high school and collegiate players. Well-tuned neural based QA systems (Yamada et al., 2018) can further improve over baselines and defeat teams of expert humans in live Quizbowl events.

However, some Quizbowl questions are fiendishly difficult for computers. Many questions have complicated coreference patterns (Guha et al., 2015), require reasoning across multiple types of knowledge, or involve wordplay. Given that models are truly challenged by these difficult types of questions, how can we generate and analyze more of them?

2.2 Adversarial Question Writing

One approach to evaluate models beyond a typical test set is through adversarial examples (Szegedy et al., 2013) and other intentionally difficult inputs. However, language is hard to modify (e.g., replacing word tokens) without changing the meaning of the input. Past work side-steps this difficulty by modifying examples in a simple enough manner to preserve meaning (Jia and Liang, 2017; Belinkov and Bisk, 2018). However, it is difficult to generate complex examples that expose richer phenomena through these automatic means. Instead, we propose to use human adversaries in a process we call *adversarial writing*.

In this setting, question writers are tasked with generating *challenge questions* that break a QA system but are still answerable by humans. To facilitate this breaking process, we expose model predictions and interpretation methods to question writers through a user interface. This allows writers to see what changes should be made to confuse the model. For

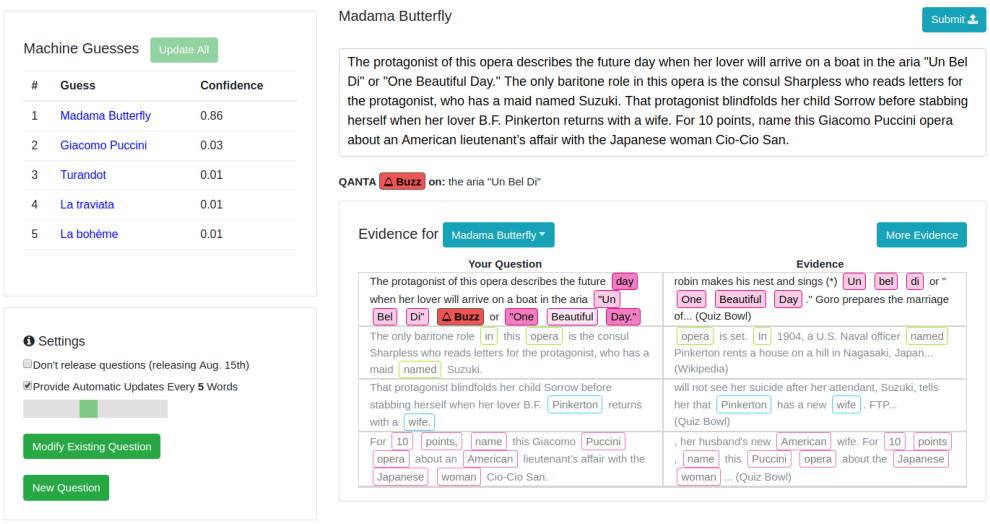

Figure 2: The writer inputs a question (top right), the system provides guesses (left), and explains why it's making those guesses (bottom right). The writer can then adapt their question to "trick" the model.

example, our interface highlights the revealing "Un Bel Di" clue in bright red.

The user interface makes the adversarial writing process interactive and model-driven, which contrasts to past human adversary settings (Ettinger et al., 2017). The result is a challenge set that explicitly exposes a model's limitations by design. While existing held-out test sets for Quizbowl provide questions that are likely to be asked in an actual *human* tournament, these challenge questions highlight rare and difficult QA phenomena that stump *computers*.

2.3 User Interface

The interface (Figure 2) provides the top five predictions (*Machine Guesses*) from a non-neural model, the baseline system from a NIPS 2017 competition that used Quizbowl as a shared task (Boyd-Graber et al., 2018). This model uses an inverted index built using the competition training data (which consists of Quizbowl questions and Wikipedia pages) and is capable of defeating advanced Quizbowl teams.

We select an information retrieval model as it enables us to extract meaningful reasoning behind the model's predictions. In particular, the Elasticsearch Highlight API (Gormley and Tong, 2015) visually highlights words in the *Evidence* section of the interface. This helps users understand which words and phrases are present in the training data and may be revealing the answer. Though the users never see outputs from a neural model, neural models still struggle (Section 4).

2.4 A Question Writing Example

To see how a player might write a question with the system, we walk through an example of writing a question sentence by sentence. The user first selects the answer to their question (restricted to entities that have Wikipedia pages). Suppose they select "Johannes Brahms" (a German Romantic composer) as their answer and begin their question with

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

The system *buzzes* correctly (i.e., it has enough information to interrupt the question and provide the correct answer) after the word composer. In the interface, the user can see the unique name "Karl Ferdinand Pohl" appears in Brahms' Wikipedia page.

The user now knows to avoid that specific phrase and rather describe Karl Ferdinand Pohl's position instead of naming him directly, rewriting the first sentence as

> 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.

The system now thinks that Frédéric Chopin is the most likely answer.

The user continues this process to create entire questions that the model cannot solve. We log information as the user writes, which lets us identify the exact types of edits that cause a model's prediction to change. We include the anonymized edit history in our dataset.

2.5 The Question Writing Community

In general NLP settings, the adversarial writing process can be crowdsourced using untrained annotators. Though, in this work, we focus on the domain of Quizbowl, where annotators must have extremely strong Trivia knowledge. In particular, we connect with question writers who craft questions for the hundreds of annual Quizbowl tournaments (Jennings, 2006). We encourage the writers to use our interface, awarding prizes to the ones whose questions were chosen to be played at a live human-computer match (Section 4.1).

2.6 Preserving Question Quality

Aside from avoiding degenerate questions (ungrammatical, uninteresting, or unfit for human consumption), we avoid questions so different from the training data that models have little chance to succeed. That is, we constrain the adversarial writing process to generate questions that are not only answerable by humans but also comparable to existing questions. Fortunately, Quizbowl has a standard question format (Lujan and Teitler, 2003): they follow a common paragraph structure, are well edited for grammar, and finish with a simple "giveaway" 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 content.

2.7 Dataset Statistics

To assemble the dataset, we first remove invalid questions through a verification process described in the subsequent section. This yields a dataset of 807 challenge questions made of 4017 sentences. 85 unique writers contributed questions and we intend to have twice-yearly competitions to continually collect data.

3 Validating Written Questions

We do not want to collect invalid questions that are a jumble of random characters or contain insufficient information to discern the answer. Thus, we first automatically filter out invalid questions based on length, the presence of vulgar statements, or repeated submissions (including re-submissions from the Quizbowl training or evaluation data). Next, we manually verify all of the resulting questions appear legitimate and that no obviously invalid questions made it into the challenge set.

Now that we believe the questions are valid, we need to confirm their *difficulty* according to human Quizbowl players. To do so, we play a portion of the submitted questions in a human-only Quizbowl event, using intermediate and expert players (current and former collegiate Quizbowl players) as the human baseline. To select which questions to play, we first group the challenge set into common Quizbowl categories (e.g., Literature, Science, Arts) and randomly sample sixty total questions to match typical tournament distributions.[2] As a baseline, we additionally select sixty unreleased high school tournament questions (to ensure no player has seen them before). These are sampled with the same category distribution.

When answering in Quizbowl, a player must interrupt the question with a buzz. The earlier that a player buzzes, the less of a chance their opponent has to answer the question before them. To capture this, we consider two metrics to evaluate performance, the average buzz position (as a percentage of the question seen) and the corresponding answer accuracy. We randomly shuffle the baseline and challenge questions, play them, and record these two metrics. On average for the challenge set, humans buzz with 41.6% of the question remaining and an accuracy of 89.7%. On the baseline 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

[2] https://www.naqt.com/hs/distribution.jsp

earlier for the challenge questions (a two-sided t-test, $p = 0.0047$). Humans find the challenge questions *easier* on average than the regular test examples (they buzz much earlier). We expect human performance to be comparable on the questions not played, as all questions went through the same submission and post-processing.

4 Models and Experiments

This section evaluates QA systems on the challenge questions. We consider diverse models: ones based on recurrent networks, feed-forward networks, and information retrieval systems to explore the difficulty of the examples.

We consider two neural models: a recurrent neural network (RNN) and Deep Averaging Network (Iyyer et al., 2015, DAN). The two models treat the problem as text classification and predict which of the answer entities the question is about. The RNN is a bidirectional GRU (Cho et al., 2014) and the DAN uses fully connected layers with a word vector average as input.

To train the systems, we use the data shared in the 2017 NIPS Human-Computer Question Answering competition (Boyd-Graber et al., 2018). The dataset consists of about 70,000 questions with 13,500 answer options. We split the data into validation and test sets to provide baseline evaluations for the models. We also report results on the baseline information retrieval system (IR) shown to users during the writing process. For evaluation, we report the accuracy as a function of the question position. The accuracy varies as the words are fed in (mostly improving, but occasionally degrading).

The buzz position of all models significantly degrades on the challenge set. We compare the accuracy on the original test set (*Test Questions*) to the challenge questions (*Challenge Questions*) in Figure 3.

For both the challenge and original test data, the questions begin with abstract clues that are difficult to answer (accuracy at or below 10%). However, during the crucial middle portions of the questions (after revealing 25% to 75%), where buzzes in Quizbowl matches most frequently occur, the accuracy on original test questions rises significantly quicker than the challenge ones. For both questions, the accuracy rises towards the end as the clues get simpler and become "give-aways".

The average human results are displayed on the left of Figure 3 and show a different trend. For both question types, human accuracy rises very quickly after about 50% of the question. We suspect this occurs because the "give-aways", which often contain common sense or simple knowledge clues, are easy for humans but quite difficult for computers. The reverse is true for the early clues. They contain quotes and rare entities that models can retrieve but humans struggle to remember. We further contrast human and machine performance in a live event (Section 4.1) and additionally show human results grouped by ability in the Appendix.

The two neural models decreased more in absolute accuracy than the IR system. This is quite surprising, as users never observe the output of a neural system at any time during the writing process. The DAN model had the largest absolute accuracy decrease (from 54.1% to 32.4% on the full question), which we suspect is due to the vector average of input embeddings not capturing the difficult wording of the challenge questions. Another explanation for the degradation in model performance is a lack of training examples that contain the specific linguistic phenomena present in the challenge questions. This suspicion may help explain the large drop in neural model accuracy, as neural models may be more sensitive to changes in the data distribution than an IR system. Section 5 explores what linguistic features of the questions fool computers.

4.1 Humans Dominate Live Exhibition

To further explore the effect of the challenge questions on both human and computer performance, we host an in-person, live human-computer exhibition match. We organize a human team consisting of national Quizbowl experts and pit them against the current state of the art Studio Ousia system (Yamada et al., 2018). This model was the best computer system at the 2017 NIPS shared task. At the live NIPS event, this model defeated a team of expert humans by a score of 475–200 when playing on unreleased tournament questions from the shared task test set.

The Studio Ousia system works by aggregating answer prediction scores produced 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, govern-

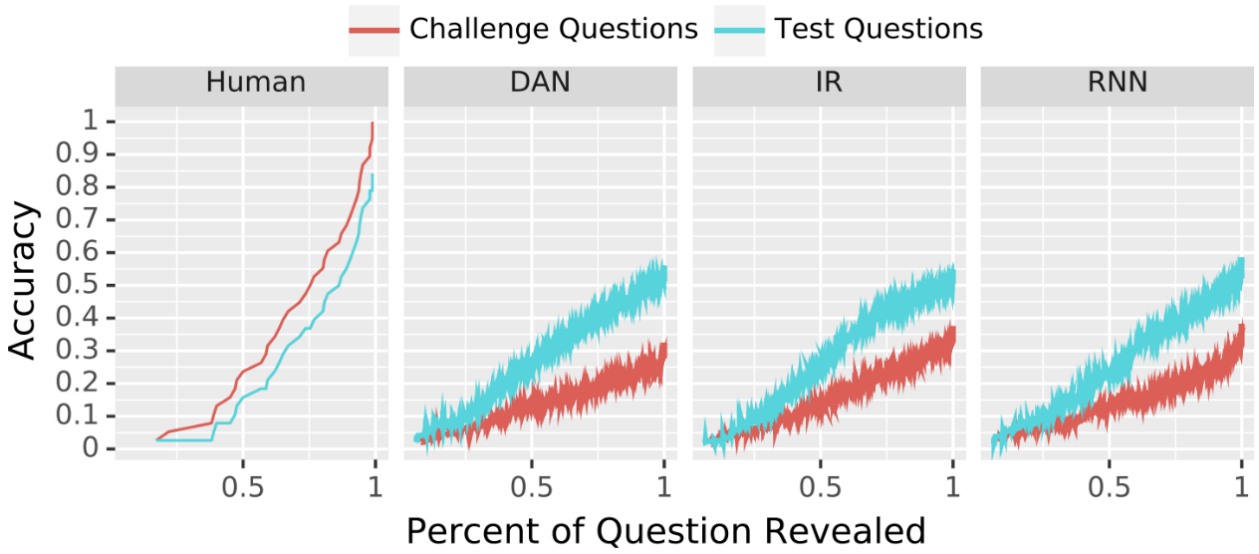

Figure 3: Both types of questions begin with abstract clues the models are unable to capture, but the challenge questions are significantly harder during the crucial middle portions (0.25 to 0.75) of the question. The human results (left) are on a smaller sample of challenge questions and on a different (but similar) set of test questions. The noise in the right three plots is due to length normalization and a moving average.

ment 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 Freebase [3] and using word vectors that incorporate information from Wikipedia's knowledge graph. As the demonstration is live, there is only one opportunity for the system to decide when to buzz.

The live event followed a standard Quizbowl tossup-only format, using forty questions selected to match typical tournament distributions. Despite the Studio Ousia system displaying superhuman ability in the past (defeating a national class team at NIPS), the human team won handily by a score of 300–30. [4] These results corroborate the findings of the previous experiments: the challenge questions are harder for machines, but easier for humans.

On the "regular" questions of the NIPS live event, the Studio Ousia system had an uncanny ability to buzz early, reflecting the presence of memorizable clues the model could exploit. Though, when playing on the challenge questions, the model buzzed significantly later, degrading from world-class performance to being defeated as if it was a novice. On incorrect guesses, the model frequently gave ludicrous answers, for example providing a poem "The Charge of the Light Brigade" when given an entire question about the number 450. To understand why the challenge questions are so difficult, we analyze their linguistic properties in the subsequent section.

5 Challenge Set Reveals Model Limitations

This section analyzes the challenge questions to understand the source of their difficulty. We harvest recurring patterns using the user edit logs and corresponding model predictions, grouping the questions into linguistically-motivated clusters (Table 1). We use 100 question sample to determine the relative frequency of each linguistic cluster, double counting questions into multiple clusters when necessary. These groups reveal common model errors from diverse phenomena.

A portion of the examples contain clues that are unseen during training time. These clues trivially break systems and result from a lack of data rather than a lack of language understanding. For example, the common knowledge clue "this man is on the One Dollar Bill". However, because we experiment with systems that are not able to capture unseen, open-domain information, we do not investigate these examples

[3] https://developers.google.com/freebase/
[4] https://youtu.be/wf5GHwDb6Ig

further. 26% of the challenge questions consist of unseen clues.

5.1 Clues Written in A Misleading Manner

The first categories of challenge questions contain previously seen clues that have been written in a misleading manner. Table 1 shows snippets of exemplary challenge questions for each category.

Paraphrases A common adversarial writing strategy (38% frequency) is to paraphrase clues to remove exact n-gram matches from the training data. This renders an IR system useless but also hurts neural models.

Entity Type Distractors Whether explicit or implicit in a model, one key component for QA is determining the answer type that is desired from the question. Writers take advantage of this by providing clues that lead the system into selecting a wrong answer type. For example, in the second question of Table 1, the "lead in" implies the answer may be an actor. This triggers the RNN model to answer Don Cheadle despite previously seeing the Bill Clinton "playing a saxophone" clue. These distractors occur in 7% of the challenge questions.

5.2 Composing Existing Knowledge

The other categories of challenge questions require composing knowledge from multiple existing clues. Snippets of exemplary challenge questions are shown in Table 2.

Triangulation In these challenge questions, several entities that have a first-order relationship to the correct answer are given. The system must then triangulate the correct answer by "filling in the blank". For example, in the first question of Table 2, the place of death of the entity is given. The training data contains a clue about the place of death (The Battle of the Thames) reading "though stiff fighting came from their Native American allies under Tecumseh, who died at this battle". The system must connect these two clues to answer. 15% of the challenge questions contain Triangulation.

Operator One extremely difficult question type requires applying a mathematical or logical operator to the text. For example, the training data contains a clue about the Battle of Thermopylae reading "King Leonidas and 300 Spartans died at the hands of the Persians" and the second question in Table 2 requires one to add 150 to the number of Spartans. This question caused the ludicrous error discussed in Section 4.1. Another example of an operator question is "This man is not Sherlock Holmes, but he lives at 221B Baker Street" (Dr. Gregory House). 5% of the challenge questions contain Operator clues.

Multi-Step Reasoning The final question type requires a model to make multiple reasoning steps between entities. For example, in the last question of Table 2, a model needs to make a step first from the "I Have A Dream" speech to the Lincoln Memorial and an additional step to reach President Abraham Lincoln. This phenomenon occurs in 25% of the challenge questions.

5.3 Suggestions to Improve QA Models

On traditional Quizbowl questions, the models defeat top human teams. The challenge questions (which highlight particularly difficult or rare aspects of QA) show that computer-assisted annotators can craft intriguing, linguistically-motivated examples that models are incapable of handling. While we cannot precisely blame the degradation in performance on the model or the training data, the challenge questions provide clues how we can improve both aspects of QA systems.

The first set of challenge questions (Section 5.1) are either difficult to parse or are complex paraphrases of existing clues. Models may become invariant to these examples if more of them are integrated into the training process as a form of data augmentation. Perhaps a system that can adversarially paraphrase questions could help generate these types of examples. In particular, Iyyer et al. (2018) create a paraphrase system that increases the robustness of a natural language inference model.

Another research direction is to create models that are capable of conducting multi-step inference across disjoint documents (as is required in the Abraham Lincoln question in Table 2). The WikiHOP dataset (Welbl et al., 2017) contains supervision for seeking and combining pieces of evidence. Transferring models from WikiHOP to other datasets (such as Quizbowl) can take advantage of this additional supervision that may not be present in the target task.

Set	Question	Prediction	Rationale
Training	Name this sociological phenomenon, the *taking of one's own life*.	Suicide	Paraphrase
Challenge	Name this *self-inflicted method of death*.	Arthur Miller	
Training	Clinton played the *saxophone on The Arsenio Hall Show*	Bill Clinton	Entity Type Distractor
Challenge	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	

Table 1: Snippets from challenge questions show the difficulty in retrieving previously seen evidence. *Training* questions indicate relevant snippets from the training data. *Answer* displays the RNN model's answer prediction (always correct on Training, always incorrect on Challenge).

Question	Prediction	Answer	Rationale
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	Triangulation
This number is one hundred and fifty more than the number of Spartans at Thermopylae.	Battle of Thermopylae	450	Operator
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 2: Snippets from challenge questions show examples of composing existing evidence. *Answer* displays the correct answer (all models were far from correct). For these examples, connecting the training and challenge clues is quite simple for humans but very difficult for models.

Though Wikipedia and Quizbowl are information-rich, they do not contain all the information necessary to answer every question. QA systems that integrate a knowledge base end-to-end (Bordes et al., 2014) or use web queries (Chen et al., 2017) could better answer questions with unseen clues.

6 Dicussion and Future Work

Our adversarial writing process uncovers model limitations using a human-in-the-loop. Medium-scale implementations of this process, such as our Quizbowl setting, do not need to produce a dataset large enough for training. Rather, the data needs to be sufficiently large for linguistic analyses to identify recurring model failure patterns. These type of failures may occur in the original test data but will do so at a significantly lower frequency. Additionally, challenge datasets are especially useful for NLP tasks where accuracies on existing test sets are extremely high.

If significant resources are available, our adversarial writing setting can yield high-quality *training* datasets. For general NLP tasks such as textual entailment, we can present model interpretations and predictions to aid in the generation of challenging examples. This new annotation method is salient given the difficulty of collecting large-scale datasets that do not contain superficial clues models can use to "game" a task (Gururangan et al., 2018; Chen et al., 2016). Our adversarial writing framework alleviates these annotation artifacts by exposing model pathologies (and their learned artifacts) *during* the data annotation process.

Our adversarial writing setup requires clear interpretations of a model. This drove our use of an IR

system rather than a neural one. In future work, we hope to incorporate neural interpretations (Montavon et al., 2017). Fortunately, annotators can still generate challenging examples for neural systems even using IR output. The effort required from annotators increases during the adversarial writing process, as they may need to rewrite an example numerous times. However, better model interpretation techniques and visualizations can ease this burden.

Another benefit of leveraging human adversaries is that they can generate examples that are more diverse than automatic methods (Jia and Liang, 2017; Iyyer et al., 2018). This diversity provides insight into numerous model limitations, rather than a single one. Combining these two approaches, perhaps by training models to mimic user edit history, could be fruitful to explore in future work.

7 Related Work

Creating evaluation datasets to get a fine-grained analysis of particular linguistics features or model attributes has been explored in past work. The LAMBADA dataset tests a model's ability to understand the broad contexts present in book passages (Paperno et al., 2016). Other work focuses on natural language inference, where challenge examples highlight existing model failures (Wang et al., 2018; Glockner et al., 2018; Naik et al., 2018). Our work is unique in that we use human as adversaries to expose model weaknesses, which provides a diverse set of phenomena (from paraphrases to multi-hop reasoning) that models can't solve.

Other work has explored specific limitations of NLP systems. Rimell et al. (2009) show that parsers struggle on test examples with unbounded dependencies. The most closely related work to ours is Ettinger et al. (2017) who also use humans as adversaries. Unlike their Build-it Break-it setting, we have a ready-made audience of "breakers" who are motivated and capable of generating adversarial examples. Our work also differs in that we use model interpretation methods to facilitate the breaking in an interactive manner.

Finally, other methods have found very simple input modifications can break neural models. For example, adding character level noise drastically reduces machine translation quality (Belinkov and Bisk, 2018), while paraphrases can fool textual entailment and visual question answering systems (Iyyer et al., 2018; Ribeiro et al., 2018). Jia et al. (2017) place distracting sentences at the end of paragraphs and cause QA systems to incorrectly pick up on the misleading information. These types of input modifications can evaluate one specific type of phenomenon and are complementary to our approach.

8 Conclusion

It is difficult to automatically expose the limitations of a machine learning system, especially when that system solves a fixed held-out evaluation set. In our setup, humans try to break a trained system. By supporting this breaking process with interpretation methods, users can understand what a model is doing and how to create challenging examples for it. An analysis of the resulting data can reveal unknown model limitations and provide insight into how and when machine learning systems work.

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*.

Antoine Bordes, Sumit Chopra, and Jason Weston. 2014. Question answering with subgraph embeddings. In *Proceedings of Empirical Methods in Natural Language Processing*.

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

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*.

Danqi Chen, Adam Fisch, Jason Weston, and Antoine Bordes. 2017. Reading wikipedia to answer open-domain questions. In *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*.

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*.

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*.

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. September.

Mandar Joshi, Eunsol Choi, Daniel S. Weld, and Luke Zettlemoyer. 2017. Triviaqa: A large scale distantly supervised challenge dataset for reading comprehension. In *Proceedings of the Association for Computational Linguistics*.

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

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

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*.

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*.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. Squad: 100,000+ questions for machine comprehension of text. In *Proceedings of Empirical Methods in Natural Language Processing*.

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*.

Laura Rimell, Stephen Clark, and Mark Steedman. 2009. Unbounded dependency recovery for parser evaluation. In *Proceedings of Empirical Methods in Natural Language Processing*.

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*.

Alex Wang, Amapreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R. Bowman. 2018. Glue: A multi-task benchmark and analysis platform for natural language understanding. *arXiv preprint arXiv:1804.05922*.

Johannes Welbl, Pontus Stenetorp, and Sebastian Riedel. 2017. Constructing Datasets for Multi-hop Reading Comprehension Across Documents. In *Transactions of the Association for Computational Linguistics*.

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*.

A Appendix A: Human Results by Ability

We separate the results of the three different groups of human players (*Dilettante*, *Expert*, and *National* class) in Figure 4. The three groups played on different but overlapping subsets of the challenge questions. The intermediate and expert teams did not face an opponent, while the national class team faced off against the Studio Ousia model. The national class team had the pressure of playing on stage with a few hundred audience members.

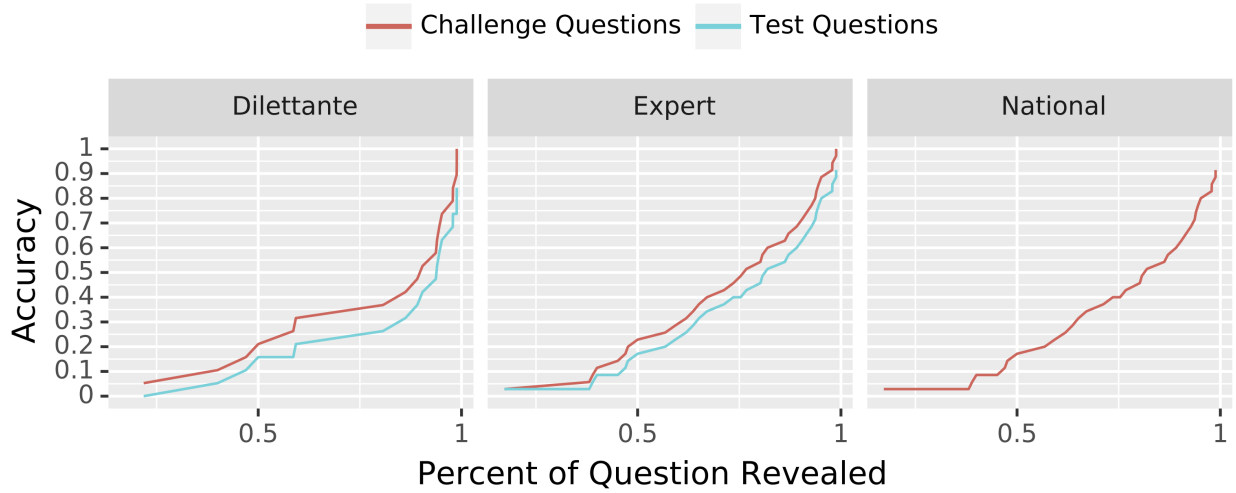

Figure 4: The three different groups of human players (dilettante, expert, and national class) are shown. Each team played on overlapping subsets of the challenge questions in a live setting. For all three human groups, the accuracy steadily rises as clues are revealed and then quickly jumps when the "give-away" clues arrive.