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

Eric Wallace
University of Maryland, College Park
ewallac2@umd.edu

Jordan Boyd-Graber
University of Maryland, College Park
jbg@umiacs.umd.edu

Abstract

Modern question answering systems have been touted as approaching human performance. However, existing question answering datasets are imperfect tests. Questions are written with humans in mind, not computers, and often do not properly expose model limitations. To address this, we develop an adversarial writing setting, where humans interact with trained models and try to break them. This annotation process yields a challenge set, which despite being easy for trivia players to answer, systematically stumps automated question answering systems. Diagnosing model errors on the evaluation data provides actionable insights to explore in developing 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.[1] Datasets such as SQuAD and TriviaQA (Rajpurkar et al., 2016; Joshi et al., 2017) have certainly advanced the state of the art, but are they providing the right examples to measure how well machines can answer questions?

Many of the existing question answering datasets are written and evaluated with humans in mind, not computers. Though the way computers solve NLP tasks is fundamentally different than humans. They train on hundreds of thousands of questions, rather than looking at small groups of them 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 crowd-sourced annotation process (Gururangan et al., 2018). Additionally, because existing test sets do not provide specific diagnostic information for improving models, it can be difficult to get proper insight into a system's capabilities or its limitations. Unfortunately, when rigorous evaluations are not performed, strikingly simple model limitations can be overlooked (Belinkov and Bisk, 2018; Ettinger et al., 2017; Jia and Liang, 2017).

To address this lacuna, we ask trivia enthusiasts—who write new questions for scholastic and open circuit tournaments—to create examples that specifically challenge Question Answering (QA) systems. We develop a user interface (Section 2) that allows question writers to adversarially craft these questions. This interface provides a model's predictions and its evidence from the training data to facilitate a model-driven annotation process.

Humans find the resulting challenge questions *easier* than regular questions (Section 3), but strong QA models struggle (Section 4). Unlike many existing QA test sets, our questions highlight specific phenomena that humans can capture but machines cannot (Section 5). We release our QA challenge set to better evaluate models and systematically improve them.[2]

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.

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

[2] www.qanta.org

Proceedings of ACL 2018, Student Research Workshop, pages 127–133
Melbourne, Australia, July 15 - 20, 2018. ©2018 Association for Computational Linguistics

> 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 Quiz Bowl question. The question becomes progressively easier to answer later on; thus, more knowledgeable players can answer after hearing fewer clues.

2.1 The Quiz Bowl Community: Writers of Questions

The "gold standard" of academic competitions between universities and high schools is Quiz Bowl. Unlike other question answering formats such as Jeopardy! or TriviaQA (Joshi et al., 2017), Quiz Bowl questions are designed to be interrupted. This allows more knowledgeable players to "buzz in" before their opponent knows the answer. This style of play requires questions to be structured "pyramidally": questions start with difficult clues and get progressively easier (Figure 1).

However, like most existing QA datasets, Quiz Bowl 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 give small improvements over the baselines and have even defeated teams of expert humans in live Quiz Bowl events.

Although, other types of Quiz Bowl 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 these difficult types of question truly challenge models, how can we generate and analyze more of them?

2.2 Adversarial Question Writing

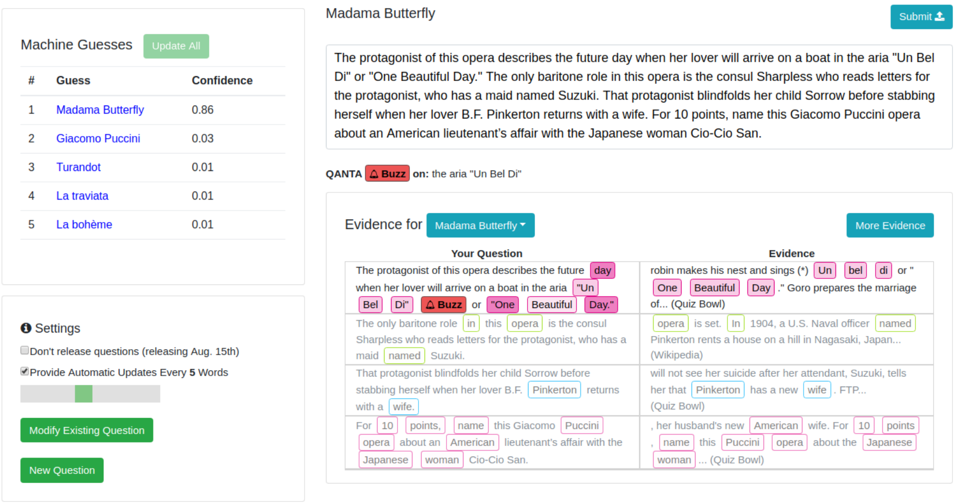
One approach to evaluate models beyond a typical test set is through adversarial examples (Szegedy et al., 2013) and other types of intentionally difficult inputs. However, language data is hard to modify (e.g., replacing word tokens) without changing the meaning of the input. Past work side-steps this difficulty by modifying examples in a simple enough manner to preserve meaning (Jia and Liang, 2017; Belinkov and Bisk, 2018). Though it is hard to generate complex examples that expose richer phenomena through 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 existing QA systems 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 system and visualize the resulting effects. For example, our system highlights the revealing "Un Bel Di" clue in bright red.

This results in a small, model-driven challenge set that is explicitly designed to expose a model's limitations. While the regular held-out test set for Quiz Bowl provides questions that are likely to be asked in an actual tournament, these challenge questions highlight rare and difficult QA phenomena that models can't handle.

2.3 User Interface

The interface (Figure 2) provides the top five predictions (Guesses) from a simple non-neural model, the baseline system from a NIPS 2017 competition that used Quiz Bowl as a shared task (Boyd-Graber et al., 2018). This model uses



Machine Guesses

#	Guess	Confidence
1	Madama Butterfly	0.86
2	Giacomo Puccini	0.03
3	Turandot	0.01
4	La traviata	0.01
5	La bohème	0.01

Settings

□ Don't release questions (releasing Aug. 15th)

☑ Provide Automatic Updates Every **5** Words

Madama Butterfly

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 lieutenant's affair with the Japanese woman Cio-Cio San.

QANTA Buzz **on:** the aria "Un Bel Di"

Evidence for Madama Butterfly

Your Question	Evidence
The protagonist of this opera describes the future day when her lover will arrive on a boat in the aria "Un Bel Di" Buzz or "One Beautiful Day."	robin makes his nest and sings (*) Un bel di or " One Beautiful Day ." Goro prepares the marriage of... (Quiz Bowl)
The only baritone role in this opera is the consul Sharpless who reads letters for the protagonist, who has a maid named Suzuki.	opera is set. In 1904, a U.S. Naval officer named Pinkerton rents a house on a hill in Nagasaki, Japan... (Wikipedia)
That protagonist blindfolds her child Sorrow before stabbing herself when her lover B.F. Pinkerton returns with a wife.	will not see her suicide after her attendant, Suzuki, tells her that Pinkerton has a new wife . FTP... (Quiz Bowl)
For 10 points, name this Giacomo Puccini opera about an American lieutenant's affair with the Japanese woman Cio-Cio San.	, her husband's new American wife. For 10 points , name this Puccini opera about the Japanese woman ... (Quiz Bowl)

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.

an inverted index built using the shared task training data (which consists of Quiz Bowl questions and Wikipedia pages).

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 system, the questions they write are quite challenging for them (Section 4).

2.4 Farmed from the Maddeningly Smart Crowd

In contrast to crowd-sourced datasets, our data comes from the fecund pool of thousands of questions written annually for Quiz Bowl tournaments (Jennings, 2006). We connect with the question writers of these tournaments, who find the adversarial writing process useful to help write high quality, original questions.

The current dataset (we intend to have twice-yearly competitions to continually collect data) consists of 651 questions made of 3219 sentences. A few of the writers have indicated that they plan to use their question submissions in an upcoming tournament. We will release these questions after those respective tournaments have completed.

3 Validating Written Questions

We next verify the validity of the challenge questions. We do not want to collect 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 Quiz Bowl 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.

We further wish to investigate not only the validity, but also the *difficulty* of the challenge questions according to human Quiz Bowl players. To do so, we play a portion of the submitted questions in a live Quiz Bowl event, using intermediate and expert players (current and former collegiate Quiz Bowl players) as the human baseline. We sample 60 challenge questions from categories that match typical tournament distributions. As a baseline, we additionally select 60 unreleased high school tournament questions (to ensure no player has seen them before).

When answering in Quiz Bowl, 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 significantly earlier for the challenge questions (a two-sided t-test yields $p = 0.0047$). 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 stages.

4 Models and Experiments

In this section, we evaluate numerous QA systems on the challenge questions. We consider a diverse set of models: ones based on recurrent networks, feed-forward networks, and IR systems to properly 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 collect the data used at 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 system (IR) shown to users during the writing process. For evaluation, we report the accuracy as a function of the question position (to capture the incremental nature of the game). 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 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 Quiz Bowl 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 "give-away" clues arrive. Despite users never observing the output of a neural system, the two neural models decreased more in absolute accuracy than the IR system. The DAN model had the largest absolute accuracy decrease (from 54.1% to 32.4% on the full question), likely because a vector average isn't capable of capturing the difficult wording of the challenge questions.

The 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 has been seen. 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 entities that models can retrieve but humans struggle to remember.

5 Challenge Set Reveals Model Limitations

In this section, we conduct an analysis of the challenge questions to better 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).

These groups provide an informative analysis of model errors on a diverse set of phenomena. In our dataset, we additionally provide the most similar training questions for each challenge question.

A portion of the examples contain clues that are unseen during training time. Many of these clues are quite interesting, 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 open-domain information, we do not investigate these examples further as they trivially break systems.

5.1 Understanding Changes in Language

The first categories of challenge questions contain previously seen clues that have been written in a misleading manner. Table 1 shows snippets of ex-

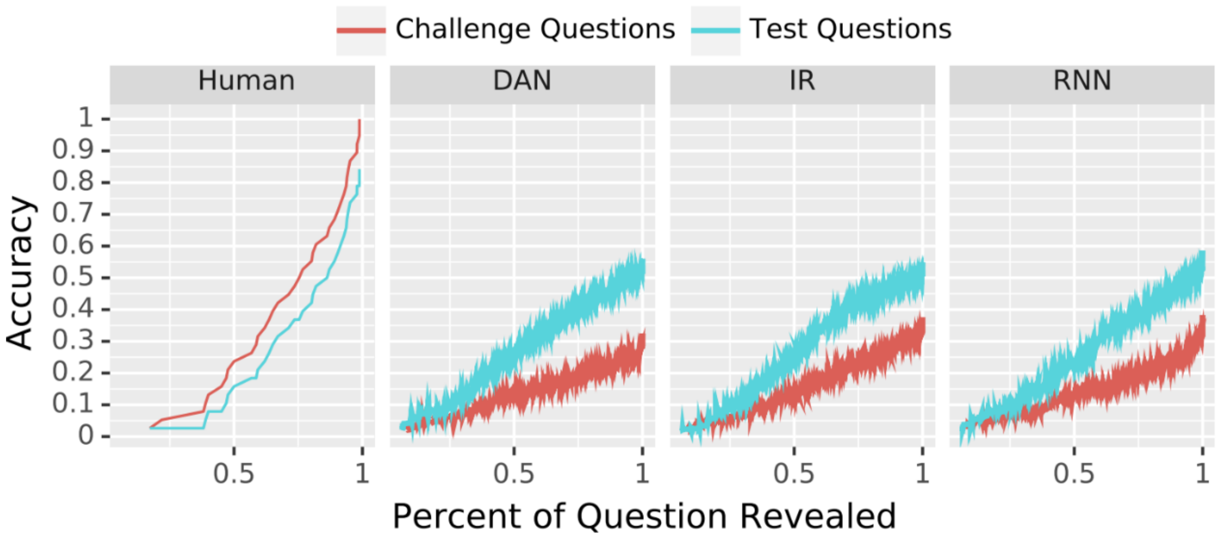


Figure 3: Both types of questions (challenge questions and original test set questions) begin with abstract clues the models are unable to capture, but the challenge questions are significantly harder during the crucial middle portions (0.25 to 0.75) of the question. The human results (displayed on the left of the figure) show their performance on a sample of the challenge questions and a separate test set.

emplar challenge questions for each category.

Paraphrases A common adversarial writing strategy 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 One key component for QA is determining the answer type that is desired from the question. Writers often 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 model to answer Don Cheadle despite previously seeing the Bill Clinton "Saxophone" clue.

5.2 Composing Existing Knowledge

The other categories of challenge questions require composing knowledge from multiple existing clues. Table 2 shows snippets of exemplar challenge questions for each category.

Triangulation In these questions, 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 and the brother of the entity are given. The training data contains a clue about the place of death (The Battle of 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.

Operator One extremely difficulty 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.

Multi-Step Reasoning The final type of questions 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 reasoning step first from the "I Have A Dream" speech to the Lincoln Memorial and an additional step to reach president Abraham Lincoln.

6 Related Work

Creating evaluation datasets to get 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). Linzen et al. (2016) create a dataset to evaluate if language models can learn subject-verb number agreement. The most closely

Set	Question	Answer	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 Reader'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 RNN Reader's answer prediction. For these examples, connecting the training and challenge clues is quite simple for humans but very difficult for models.

related work to ours is Ettinger et al. (2017) who also consider using humans as adversaries. Our work differs in that we use model interpretation methods to facilitate breaking a specific system.

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 natural language inference systems (Iyyer et al., 2018). Jia and Liang (2017) placed distracting sentences at the end of paragraphs and caused 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.

7 Conclusion

It is difficult to automatically expose the limitations of a machine learning system, especially when that system reaches very high performance metrics on a held-out evaluation set. To address this, we have introduced a human driven evaluation setting, where users try to break a trained system. By facilitating this 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 improving a system.

Acknowledgments

We thank all of the Quiz Bowl players and writers who helped make this work possible. We also thank the anonymous reviewers and members of the UMD "Feet Thinking" group for helpful comments. JBG is supported by NSF Grant IIS-1652666. Any opinions, findings, conclusions, or recommendations expressed here are those of the authors and do not necessarily reflect the view of the sponsor.

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*.

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* .

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 .

Allyson Ettinger, Sudha Rao, Hal Daumé III, and Emily M. Bender. 2017. Towards linguistically generalizable nlp systems: A workshop and shared task .

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 .

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. http://books.google.com/books?id=tXi_rfgUWCcC.

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

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*.

Tal Linzen, Emmanuel Dupoux, and Yoav Goldberg. 2016. Assessing the ability of lstms to learn syntax-sensitive dependencies 4:521–535.

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 .

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. Squad: 100,000+ questions for machine comprehension of text .

Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian J. Goodfellow, and Rob Fergus. 2013. Intriguing properties of neural networks .

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