Jordan Boyd-Graber, Shi Feng, and Pedro Rodriguez. **Human-Computer Question Answering: The Case for Quizbowl**. *The NIPS '17 Competition: Building Intelligent Systems*, 2018, 10 pages.


```
    @inbook{Boyd-Graber:Feng:Rodriguez-2018,
Publisher = {Springer Verlag},
Author = {Jordan Boyd-Graber and Shi Feng and Pedro Rodriguez},
Url = {docs/2018_nips_qbcomp.pdf},
Booktitle = {The NIPS '17 Competition: Building Intelligent Systems},
Title = {Human-Computer Question Answering: The Case for Quizbowl},
Year = {2018},
Editor = {Sergio Escalera and Markus Weimer},
}
```

Chapter 1
Human-Computer Question Answering: The Case for Quizbowl

Jordan Boyd-Graber, Shi Feng, Pedro Rodriguez
{jbg,shifeng,pedro}@cs.umd.edu
University of Maryland

1.1 What is quizbowl?

Quizbowl is a competition played between middle schools, high schools, and colleges across the English-speaking world. It sometimes takes different names—in the UK, it's called "University Challenge—but the central idea is the same: questions that test academic knowledge are read to teams. Those teams interrupt the question when they know the answer. These games are fun, engaging, and help the players prove and improve their knowledge of the liberal arts, science, and broader culture.

There are many games that test knowledge from game shows to pub quizzes, but there is a key component that makes quizbowl unique and special: **questions are specially written so that they can be interrupted**. Other question answering frameworks focus on single sentence questions. If both players or neither players can answer the question, then the question is useless in determining who knows more. Thus, you need many more questions of wildly varying difficulty to figure out who knows more. Quizbowl rewards different levels of knowledge in the *same question*, making it more efficient (and fun).

To see how this works, take the example question in Figure 1.1; it begins with obscure information and then gets to trivial wordplay. Players with deep knowledge of music history and the libretto can answer early. Players who have memorized the names of characters can answer midway through, and players who just know titles of Mozart operas can get it at the end through wordplay. Because players can interrupt the question as its being read, it is a more effective way of distinguishing who knows more.

1.2 Why should this be a computer competition?

We haven't yet mentioned computers. We have only argued that quizbowl is a fun game and an efficient way to figure out who know more about a subject. Why we should have computers playing quizbowl?

In this section, we briefly describe our rationale for using quizbowl as a question answering task before describing our human-computer question answering competition at NIPS 2017.

1.2.1 Who's Smarter?

The deluge of computer question answering systems and datasets shows that there is keen interest in knowing which computer systems are smartest and can learn most effectively from reading the web. However, most of the computer science community has ignored the lessons learned from decades (centuries?) of asking questions of humans to figure out who is smarter or a better learner. For the same reasons that quizbowl is the "gold standard" of knowledge-based competitions in humans, we should adopt the same lessons for computer-based question answering.

At its premiere, the librettist of this opera portrayed [a character who asks for a glass of wine with his dying wish]$_1$. [That character]$_1$ in this opera is instructed to ring some bells to summon his love. At its beginning, [a man]$_2$ who claims to have killed a serpent has a padlock put on [his]$_2$ mouth because of [his]$_2$ lying. The plot of this opera concerns a series of tests that Tamino $_2$ must undergo to rescue Tamina from Sorastro. For 10 points, name this Wolfgang Mozart opera titled for an enchanted woodwind instrument.
ANSWER: The Magic Flute

Fig. 1.1: An example quizbowl question. The question starts with difficult clues and gets easier through the course of the question. Solving the answer to the question requires deep knowledge (e.g., that Emanuel Schikaneder portrayed Papageno in the premiere even though neither is mentioned by name). Two coreference groups corresponding to Papageno (1) and Tamino (2) are highlighted to show the difficulty of aligning these mentions.

1.2.2 Machine Learning

The question answering process that allows competitors to buzz after every word makes the problem more interesting from a machine learning perspective. This means that the system needs to decide after every word whether it has enough information or not to answer the question. If the system decides that it does have enough information, it can buzz.

This machine learning challenge is similar to simultaneous interpretation [1, 3]: for example, translating from German to English one word at a time. Thus, a full solution needs reinforcement learning [4] to learn how to adjust to the strategies of your opponent. However, a solution can be much simpler (Section 1.3).

1.2.3 Natural Language Processing

Let's return to the question in Figure 1.1. Answering the question correctly requires extensive coreference resolution: recognizing that characters aren't explicitly mentioned by name (Schikaneder) or that pronouns can appear before the reference (Tamino). Moreover, quizbowl represents a super-difficult version of coreference [2]: if a system can do well solving these coreference tasks, it can do well just about anywhere.

1.3 A Simple System to Scaffold Users

This section describes a very simple system we developed for users to use as a starting point in creating their submissions.[1] There are two primary components to the system: a *guesser* that generates candidates to answer and a *buzzer* that decides when to take that as the answer. Unlike other question answering scenarios where the only challenge is to generate answers, quizbowl also requires players to select how much information was needed to answer the question.

1.3.1 Guesser

Our simple guesser is based on ElasticSearch [**?**], a search engine that uses TF-IDF keyword matching. Given a query (part of the question), the system compares it with previously asked quizbowl questions, and sorts the corresponding entities based on the similarity. Our simple guesser was trained on 31862 quizbowl questions, which covers 6991 distinct entities.

[1] https://github.com/pinafore/qb-api

C: I'm User 1. I'd like to play!
S: Hi, User 1! Available questions are `[1,2,3,4]`
C: I'd like to hear Word 1 of Question 1
S: It's `Extremism`
C: I'd like to hear Word 3 of Question 1
S: It's `in`
C: I'd like to hear Word 2 of Question 1
S: It's `the`
C: I'd like to answer Question 1 `Barry_Goldwater`
S: Got it! You've answered Question 1 at Position 3 with `Barry_Goldwater`

Fig. 1.2: Example interaction of a participant client (C) with the server (S). The client repeats the process for all of the questions in the dataset.

1.3.2 Buzzer

The ElasticSearch guesser returns with each entity a score of the similarity between the document and the query. The scores provide information about how confident the guesser is and can be used to determine when to buzz. Our simple reference system uses a simple threshold to decide when to buzz. We normalize the scores of the top 10 guesses, and buzz if and only if the top score exceed a threshold (0.3) that is roughly tuned on a small held-out dataset.

1.4 Client-Server Architecture

Now that participants have the starting position of a system, we also need to provide a submission framework. Unlike other question answering tasks where entire questions are revealed at once, the nature of quizbowl requires an interactive server.

The server waits for users to submit their systems; when participants are ready, they interact with the server, which provides questions one word at a time (Figure 1.2). The server records the answer and associates it with the last word the client saw.

1.5 Results

Our competition ran in two phases: a computer only competition and a human-computer competition. After the top computer teams were selected in the computer competition, they were then placed against a top human team.

1.5.1 Computer Results

Other than calculating the accuracies of each computer system on the evaluation dataset, we also run simulated games between each pair of them. Of the seven submissions, we run this "tournament" between the three systems that performed better than the baseline system (we also include the baseline system for comparison). We use 160 test questions from the held-out dataset for both evaluations.

1.5.1.1 Accuracy

We evaluate the accuracy of each system by comparing the guesses they provided for each question, without using the buzzing information. The numbers are shown in Table. 1.1. All selected submissions beat the baseline system in terms of end-of-question accuracy, and Ousia's system out-performed other systems by a large margin.

1.5.1.2 Simulated Games

For each pair of the system we run games simulated using the test questions. We use the regular scoring system: 10 points for a correct answer, -5 for a incorrect one if not at the end of the question. Systems are only allowed to answer after each word, and we break tie randomly. To reduce the randomness in the final score, we ran 3 games and take the average for each pair (Table 1.2).

Studio Ousia's system significantly (see contributed chapter in this collection) out-performed all other systems. We observed that the other systems appear relatively conservative when it comes to buzzing, and they often only answer the question at the end. Studio Ousia's system is superior in terms of both accuracy and buzzing capability, and is thus selected to compete with top human players.

While Lunit.io (Kim et al.) had a lower accuracy than Acelove (Chen et al.), they won in more head-to-head matches, showing that knowing *when* to buzz is just as important as determining the correct answer with buzz *with*.

System	Accuracy
OUSIA	0.85
Acelove	0.675
Lunit.io	0.6
Baseline	0.55

Table 1.1: Accuracy of the systems.

System	Points	Points	System
Acelove	60	**1220**	OUSIA
Acelove	320	**615**	Lunit.io
Acelove	**470**	400	Baseline
OUSIA	**1145**	105	Lunit.io
OUSIA	**1235**	15	Baseline
Lunit.io	**605**	75	Baseline

Table 1.2: Final points from simulated games.

1.5.2 Human-Computer Games

Fig. 1.3: Trivia experts facing off against the winner of the computer phase of the competition.

Los Angeles is an epicenter for trivia competitions and game show contestants. In preparation for the NIPS competition, we recruited strong players (including some in the NIPS community) to take part in the competition (Figure 1.3).

- Raj Dhuwalia is an educator living in the LA area who is a Jeopardy champion, a winner of 250 thousand dollars on "Who Wants to be a Millionaire", and a top-two scorer at three consecutive national quizbowl championships.
- David Farris is a mathematician who was a member of national championship teams at all three levels: high school, undergrad (Harvard), and graduate school (Berkeley). This year he won the Karnataka Quiz Association championship in Bangalore on the Rest House Crescent quizzers.
- Hidehiro Anto is an undergraduate at UCAL and was a member of 3rd place Team California at 2014 HSAPQ NASAT and top scorer at 2015 NAQT ICT junior varsity competition.
- Charles Meigs is the only player to win All-Star in Division II in two consecutive years while playing for UCLA, and led the University of Maryland to a 2008 national championship.

	First Half	Second Half	Final
Human	80	120	**200**
Computer	245	230	**475**

Table 1.3: Final scores of the first place system against a strong human team.

- Sharad Vikram is a PhD candidate at UCSD but a veteran of quizbowl teams at Torrey Pines and Berkeley, where he served as president. As author of the quizBowl Database, he is also the creator of much of the data used to train these algorithms.
- James Bradbury is a research scientist at Salesforce, where he works on question answering (among many other things). As a quizbowler, he played on top 15 teams at Thomas Jefferson High School and Stanford.

In addition, the second place computer team (Lunit.io, Kim et al.) played against the University of Maryland quizbowl team a team with some of the same members had won the Academic Competition Federation tournament the previous year.

For both games, a moderator reads questions and an assistant advances the text one word at a time, simulating the client-server framework participants used to submit questions.

1.5.2.1 First-Place Game

The competition against the first place team was slow to start.[2] The computer won the first tossup on *Les Fleur du Mal* (a poetry collection by Baudelaire), and the humans got the next question on *Bimetalism* wrong (although the computer could not get the answer at the end of the question. After another human mistake on *Newton (Surname)* (which the computer could convert), the score stood at 50 for the computer against -10 for the humans.

However, the humans began a comeback with Question 6, correctly answering a question about the programming language *SQL*. However, the computer still maintained an impressive lead at the end of the first half, 260–80. The humans had a much better second half, answering four questions in a row on mythology, popular culture, and social science. The computer also had its only incorrect answer in the second half, incorrectly answering a question about Chinese art.

Nevertheless, the computer won handily (Table 1.3), although we speculate that the humans could have done better with a little more practice playing with each other and against the computer.

1.5.2.2 Second-Place Game

The second place team (Lunit.io, Kim et al.) also played against the University of Maryland Academic Quiz Team in College Park, Maryland (without an audience). The human team decisively won this game, almost doubling the computer's score.[3] The results aren't strictly comparable, as the Maryland team obviously had different strengths (being different people) and had more experience playing with each other.

1.5.3 Are computers better at quizbowl?

In many ways, we have engineered this version of the task to be generous to computer players: answers are all Wikipedia answers, we use hard questions about easy topics, and we provide the text signal perfectly to the systems.

While the systems submitted to this competition are indeed impressive, there is substantial room for improvement in making the task more challenging for computers and would be a more realistic comparison of human vs. computer performance. We discuss how to further increase the difficulty (and scientific value) of this competition.

[2] https://youtu.be/gNWU5TKaZ2Q

[3] https://youtu.be/OkgnEUDMeug

1.5.4 Improving Participation

We did not have as many entrants in our competition as we had hoped. Some of this was our own fault: we did not have as extensive and clear documentation as we had hoped, nor did we have our example system available as quickly as we would have liked. We should have clear documentation, simple example systems, and clear points to improve the example systems easily.

Future compeitions could move to more well-known platforms suce as CodaLab[4]; while our client-server system is relatively straightforward, a visible framework with slightly more complexity might be wortwhile for the additional debugging and documentation. Integrating into a more flexible web platform could also create better presentations of games that might make the submission process more fun.

Finally, having backing in the form of a monetary prize would encourage competitions and help build visibility for the competition.

1.6 Long-Term Viability of Increasingly Difficult Competitions

1.6.1 Changing the Difficulty

We have chosen nearly the optimal level of competition for our matches: difficult questions on easy topics. If the questions were easier, humans would have a speed advantage because they would be able to process information more quickly; computers often need multiple clues to triangulate information correctly. If the questions were more difficult, there would be less redundant information available from previous questions and Wikipedia, which would expose the fragility of systems: the same information can be worded in many different ways, and most systems are not robust to these paraphrases.

1.6.2 Speech Recognition

The most obvious step to make the competition more realistic for computers is to provide a speech signal rather than a text signal.[5] In particular, infrequent, highly specific named entities (e.g., "phosphonium ylide", "Temujin", or "Techumseh") are often key clues but are very hard for existing ASR systems to perfectly detect.

1.6.3 Adversarial Question Writing

We have been using questions written in the same way that they are for human players. However, computer players have different strengths and weaknesses than computer players. By highlighting clues / phrases that are easy for a computer and asking question writers to rephrase or avoid those clues, we can focus the questions on that are truly challenging for computers to answer.

1.6.4 Bonus Questions

Quizbowl games are not just tossups; typically the team that answers a tossup correctly is rewarded with the opportunity to answer a "bonus" question. These questions are typically multipart, related questions that probe the depths of teams' knowledge (Figure 1.4). Successfully answering these questions would require both deeper knowledge and cross-sentence reasoning.

[4] https://worksheets.codalab.org/

[5] In a sense, our computers are playing quizbowl in the same way deaf students play.

In this play, Tobias jokes about his wife's concern that she may someday go mad. For 10 points each:

10 Name this play in which Tobias and Agnes allow the distressed couple Harry and Edna to stay in the bedroom of their daughter Julia, who is disturbed to find it occupied when she comes home after a divorce.
ANSWER: A Delicate Balance

10 Other odd couples in plays by this author of *A Delicate Balance* include *Zoo Story*'s Peter and Jerry, as well as George and Martha from *Who's Afraid of Virginia Woolf?*
ANSWER: Edward Albee

10 In *Who's Afraid of Virginia Woolf?*, Nick realizes this fact about George and Martha's family after George says an unseen character died trying to drive around a porcupine and Martha shouts, "you cannot do that!"

ANSWER: their alleged son does not exist *[or that George and Martha are childless; or that George and Martha never had a child; accept other answers indicating that their male child is any of the following: not real; fictional; imaginary; made up; never existed; etc.; do not accept or prompt on "their son is dead"]*

Fig. 1.4: An example of a bonus question; teams work together to answer these more difficult questions, which often require reasoning across the three individual component questions.

1.6.5 Open Domain Answers

Most (over 80%, sometimes higher depending on format) answers in quizbowl map to Wikipedia page titles. However, answers sometimes do not fit into this mold. For example, some of them are common link questions, or they are.

- *Computational math questions*: uncommon at higher competition levels, computational math questions do not map to Wikipedia pages in the same way as other questions.
- *Common link questions*: sometimes a question will ask about several entities at once, and request players respond with how they fit together ("coats" in literature, the word "journey" appearing in titles of works, "ways that Sean Bean has died in films")
- Some answers do not have Wikipedia pages because they are rare or do not fit into Wikipedia's organization (e.g., "Gargantua", a character by Rabelais)

1.7 Comparison to Other Tasks

Quizbowl is a factoid question answering tasks, among which it is unique for being pyramidal - questions consist of multiple clues arranged in descending order of difficulty, with the hardest information first and the easiest at the end. Questions are revealed word by word, and players can buzz in when they are confident to answer. As a result, quizbowl challenges a system to both effectively incoporate information on-the-fly and accurately determine when the information is sufficient. Finally since quizbowl questions are revealed word by word, computers no longer have the advantage of reading speed against human. This makes quizbowl uniquely suitable for human-computer competitions.

There are several other question answering datasets that use trivia questions. But as pyramidality is unique to quizbowl, we will compare them on some different aspects. We focus on three main attributes: answer definition, support set, and query type. Answer definition refers to both the answer space (e.g. open domain vs closed domain), and how the answer is defined (e.g. an entity vs a span in a document). Support set refers to the context that is provided for each example. Query type concerns in general how the queries are formed and is closely related to the support set.

The task most similar to quizbowl is TriviaQA [5], and we compare them in detail:

- **Answer**: Similar to Quizbowl, most of TriviaQA's answers are titles in Wikipedia. TriviaQA has 40,478 unique answers, while Quizbowl has 8,091.
- **Support**: Each Quizbowl answer has on average thirteen previously asked questions, each composed of four sentences on average. Each TriviaQA questions has on average six supporting documents collected automatically from Wikipedia and web search. Because this evidence collection process is automated, there is no guarantee that the answer will appear in the documents (79.7% from Wikipedia and 75.4% from web search). The support set of TriviaQA makes it more suitable for reading comprehension methods compared to Quizbowl.

- **Query**: TriviaQA has single sentence questions, and the clues are usually given in the form of a relationship of the entity in question with another entity, for example "who is the author of King Lear?". Quizbowl questions have four sentences on average, and start with more obscure clues. The clues, especially those from the first two sentences, almost always require high-order reasoning between entities, for example "The title character of this short story describes himself as driven off the face of the earth." Quizbowl requires more complicated reasoning, across multiple sentences, and requires broader knowledge.

Other similar tasks include: SearchQA [**?**], a dataset of *Jeopardy!* questions paired with search engine snippets as evidence; WikiQA [**?**], a dataset automatically constructed from Bing queries that focuses on answer sentence selection. We note that these tasks don't share the pyramidality feature of Quizbowl and are less suitable for human-computer competitions.

Unlike other claims of human-level ability on question answering tasks,[6] quizbowl is a computerized setting of a task that humans already do. This makes it more straightforward to make realistic comparisons of relative performance. Its inherent fun makes it exciting to use it as a lens for researchers and lay audiences alike to measure the progress of NLP and ML algorithms.

1.8 Conclusion

Quizbowl is a fun activity that is engaging to audiences and participants. It is easy to begin but difficult to master, and mastery of Quizbowl requires significant advances in natural language processing and machine learning. We hope that this initial NIPS competition provides the foundation for future competitions on a wider scale.

References

1. Alvin Grissom II, He He, Jordan Boyd-Graber, John Morgan, and Hal Daumé III. 2014. Don't Until the Final Verb Wait: Reinforcement Learning for Simultaneous Machine Translation. In *Empirical Methods in Natural Language Processing*. docs/2014_emnlp_simtrans.pdf
2. 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*.
3. He He, Jordan Boyd-Graber, and Hal Daumé III. 2016a. Interpretese vs. Translationese: The Uniqueness of Human Strategies in Simultaneous Interpretation. In *North American Association for Computational Linguistics*. docs/2016_naacl_interpretese.pdf
4. He He, Kevin Kwok, Jordan Boyd-Graber, and Hal Daumé III. 2016b. Opponent Modeling in Deep Reinforcement Learning. docs/2016_icml_opponent.pdf
5. Mandar Joshi, Eunsol Choi, Dan Weld, and Luke Zettlemoyer. 2017. TriviaQA: A Large Scale Distantly Supervised Challenge Dataset for Reading Comprehension.

[6] https://www.theverge.com/2018/1/17/16900292/ai-reading-comprehension-machines-humans