PyraModel: Model for Computationally Crafting Elite Quizbowl Questions

Arman Bolouri
abolouri@umd.edu
University of Maryland, College Park
College Park, Maryland, USA

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

Abstract

Nowadays, LLM text generation is everywhere, but unfortunately, these models often make information up. To correct this problem, we need a pipeline to generate coherent text with lots of dense facts together. We propose a method of generating text with a higher degree of correctness, interestingness, and salience by efficiently retrieving facts from Wikipedia (or other encyclopedias) and feeding this information to LLMs as a basis for their generation. By establishing such a pipeline, we can allow for both an efficient and accurate way of generating specialized text rooted in facts. Although issues will already be greatly reduced, we also incorporate Reinforcement Learning from Human Feedback (RLHF) as part of this pipeline, as humans can most easily spot problems that still exist during stages of training. However, in this paper, we focus on a specific form of text generation, developing a novel approach to computationally generating good and complex Quizbowl questions. Quizbowl is a popular, competitive trivia game in which players are presented with questions about a certain topic and must give the correct answer as quickly as possible, getting more points the quicker they can answer the given questions. Useful metrics for determining good QB questions regardless of level would be pyramidality, factual correctness, and interestingness/funness. We will go more into these later. Additionally, we conclude by performing a comparison of the quality of human-generated questions to current computer-generated ones from all of our methods. This evaluation will be done both computationally and through expert QB question-writer and player ratings/feedback.

ACM Reference Format:
Arman Bolouri and Jordan Boyd-Graber. 2023. PyraModel: Model for Computationally Crafting Elite Quizbowl Questions. In *Proceedings of Computer Science Master's Degree (Scholarly Paper).* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

Scholarly Paper, Fall 2023, UMD CS Master's

ACM ISBN 978-x-xxxx-xxxx-x/YY/MM…$15.00
https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 Introduction

1.1 Question Generation

In the modern era of LLMs, text generation often suffers from hallucinations and incorrect information. Some of those hallucinations are the result of LMs trying to be helpful or entertaining. Also, to further the problem, it is often quite hard to evaluate the text generated by these language models for both accuracy and relevancy. GPT, for example, has shown remarkable capabilities in generating human-like text, but it also often spits out fabricated information. Research in this area has been increasingly significant, as these inaccuracies can have serious implications, especially in contexts where reliable information is critical. A notable study by Ren et al. in 2021 [11] explored this phenomenon, demonstrating that LLMs can confidently generate plausible but entirely fictitious statements. This was further elaborated by Bender et al. in 2021 [1], which highlighted the ethical considerations surrounding this issue, particularly in the context of misinformation. By blatantly spreading wrong information in certain domains such as cultural ones, serious ethical concerns may arise. These studies collectively underscore the importance of handling LLM outputs with caution and the need for ongoing research to improve their reliability. We attempt to better combat hallucinations both by having a more grounded generation and by focusing on a domain used to check for hallucinations, human evaluation.

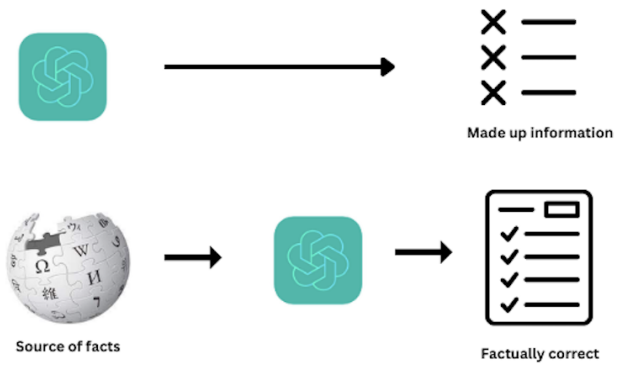

Figure 1. LLMs like GPT can make information up while generating text. However, when combined with a source of facts, they will be more likely to generate factually correct text

Thus, we propose PyraModel, a novel approach to combat hallucinations and do much more specifically tailored to generating elite QuizBowl questions. Through this technique, we focus on directly retrieving facts from Wikipedia and having LMs format such text in the format of a question. Essentially, we are using Wikipedia for fact retrieval, and existing LMs as a text generator that merges these facts into a coherent and correctly formatted question. By conditioning LLM generation on text that is proved to be correct, accuracy rises significantly. This is essentially a unique/novel form of RAG (Retrieval-Augmented Generation), where we use a separate domain to improve LLM generation. We will discuss this in more detail later in the following paper.

Ultimately, this generation domain is fact-based, has no tolerance for factual errors, and involves a community that's willing to judge the generated text for correctness (and with the innate knowledge to do so): QuizBowl players and question writers. Although we're focusing on the specific domain of question generation and particularly, QuizBowl, it is important to note that this novel fact-based generation pipeline is going to be useful for all general forms of LLM text generation.

1.2 About Quizbowl

Quizbowl is a competition where teams answer questions as soon as they can about various subjects. While past work has focused on writing adversarial examples that are difficult for machines, we will be focusing on the backbone of a quizbowl question: its pyramidal structure. Since quizbowl questions can be answered without finishing the question, it is imperative that the clues presented at the beginning of the question are more obscure and niche, whereas the clues found later are more well-known. For this reason, it is non-trivial for non-trivia nerds to create quality quizbowl questions past the high school level. Given pyramidal structure, questions are able to differentiate good players from bad players reliably. The same can be said for question-answering models and computers for the game. Essentially, generating pyramidal questions is useful as a QA metric since it inherently incorporates difficulty and obscurity.

1.3 NLP for Answering Questions

Recently, there has been an increasing interest in using NLP techniques to generate good quizbowl questions automatically for this reason as questions in datasets like SQuAD (Stanford Question Answering Dataset) do not feature this quality. More precisely, SQuAD is more designed to be used for reading comprehension (RC) in which the question is provided with a context containing the answer. Note, in this paper, we differentiate between quizbowl-esque no-context question answering (QA) and reading comprehension. Currently, the only approach that seems to be published that uses NLP to generate quizbowl questions is to use GPT4 to do such given a prompt or topic. However, although it turns out that this language model was in fact trained on ACF-style questions, it does not inherently take pyramidality into account.

1.4 Dataset

The best questions are obviously written by humans, and this takes a lot of cost and effort to do. Essentially, to minimize this, we wish to train/mimic our models on these type of questions, so we collected a dataset of human-generated quizbowl questions from the QANTA 2021 train dataset, which includes information from previous quizbowl tournaments and online question banks.

2 Related Work

2.1 Neural Approaches and SQuAD

The rise of deep learning spurred the use of models like sequence-to-sequence for question generation, as demonstrated by Serban et al. in 2016 [13]. In this vein, the SQuAD dataset [9] has been pivotal. Originally designed for question answering, SQuAD offers a vast collection of questions and answers derived from Wikipedia passages. However, this is inherently based on easier reading comprehension-based questions and non-pyramidal questions. However, this was a vital dataset for the begininnings of NLP techniques for QA.

2.2 Multi-hop Question Answering

A pronounced trend in recent question-answering research is the emphasis on multi-step, or multi-hop, reasoning. In 2018, Welbl et al. [14] presented the WikiHop dataset, constructed using Wikipedia, Wikidata, and WikiReading, promoting multi-hop reasoning across various documents. Complementing this, HotPotQA also stresses multi-hop reasoning but adopts a crowdsourcing approach for dataset creation [15]. Within quizbowl, questions on novels often necessitate multi-step reasoning, which is a main focus of our paper. A prototypical start to a question might describe a novel's character, with the answer being the author, exemplifying the multi-hop reasoning required. Extracting the initial segments of such quizbowl questions offers a reservoir of multi-hop reasoning examples that can aid in honing multi-hop reasoning models. In the world of expanding question answering datasets, Quizbowl remains distinctively unparalleled. Its genesis wasn't in machine learning research, entertainment, or natural language processing; rather, it emerged as an educational tool, driving students to learn and showcase their mastery. The Quizbowl methodology and the associated QANTA dataset have been honed over decades by brilliant writers, underscoring the art of crafting exemplary trivia questions.

2.3 Using Machines to Answer Quizbowl Questions

In 2019, Boyd-Graber et al. [10] provided a deep exploration of the incremental nature of quizbowl questions in their work "Quizbowl: The Case for Incremental Question Answering." This paper emphasized that quizbowl players have the freedom to respond at any point during the question, thereby necessitating models to interpret clues progressively and recalibrate their confidence levels in real-time. Unlike datasets tailored for specific research objectives or entertainment, Quizbowl originated as an educational instrument, aiming to foster learning and celebrate mastery over varied knowledge areas, directly corresponding to the "Manchester Paradigm". The sustained refinement of the Quizbowl methodology and the associated datasets stand as a testament to its value in the realm of trivia questions. This work also shows the key difference between how humans and many previous machine QA models play these questions: humans with building thought and strategy, and many computers with strict fact retrieval techniques (the buzzing aspect is incredibly vital for this comparison).

3 Baselines

Currently, there is not much published work on the generation of quizbowl-esque pyramidal questions. With the recent improvements in pre-trained large language models, an approach could be to create questions either by fine-tuning or prompting models such as GPT, and assessing the generated questions as baseline results. For our work, the various GPT iterations (2, 3.5, and 4) end up being the perfect baseline approach as they are currently one of the state-of-the-art tools for text generation, and are relatively easy and effortless to use. Human-written questions are going to be crafted with utmost care, precision, and accuracy, perfectly adapted to the specific difficulty level and tournament of a QB competition; thus, question writing is incredibly time-consuming and expensive. Ultimately, all this means that human-written questions are at a caliber we want to reach, and are overall a good "goal" for our approach. GPT-generated questions, on the other hand, are very convenient to generate, lending themselves to being a good baseline to compare with our "PyraModel".

One preliminary method is to use transfer learning by fine-tuning a pre-trained model such as GPT-2 [8]. In this approach, the 124M parameters GPT-2 model is fine-tunned on QANTA 2021 dataset [3]. The data is formatted as '<question text>. Answer: <answer>.' and inputted into the model. After 1000 epochs, some of the generated questions are as follows:

`A group of mercenaries fighting under a banner proclaiming "Peace, peace and peace" were called this city's "Cettines." A man in this city was killed by troops after his brother's army failed to destroy his encampment, and another of this city's rulers had been imprisoned for opposing the founding of the city. A poet in this city wrote a poem in which she sings "A high and slow and dark heart". For 10 points, name this city in the Roman Valley which had been conquered by the Romans after a successful siege by Richard of Lancaster. Answer: Alta or Alta or Alexandria.`

The question sounds like a quizbowl question, and seems to preserve the pyramidal structure at first sight, however the clues specifying events and characters are made-up.

`This poem begins with a description of a man who "can see a thousand things," who "must be in the midst of all things" and "must be loved and admired." This poem describes a man who is "or at least he [is] a true believer," who "fought for his heart," who "has the heart of a lion," and who "must be loved and admired." For 10 points, name this poem written by Joseph Conrad. Answer: On the Golden Horn or On the Golden Bough or On the Golden Leaf.`

The question seems very synthesized and the clues are not accurate. In general, questions generated by GPT-2 included inaccurate clues, disturbed the pyramidal structure by repeating similar sentences successively, and did not reflect the decreasing difficulty level towards the end of the question. It is far from being used as a tool for generating real tournament questions.

Another approach is to 'prompt-train' GPT-3.5 [2] to generate questions similar to the given examples. When the model `gpt-3.5-turbo-instruct` is few-shot prompted given 20 example questions from QANTA we get questions like the following:
`Question: This composer's final symphony includes a fourth movement in which a choir sings the words "Ode to Joy" from a poem by Friedrich Schiller. The second movement of this work begins with a somber funeral march, while the third movement features a lively scherzo. Earlier in his career, this composer wrote an opera based on a play by Goethe, titled "Fidelio." For 10 points, name this composer of the Ninth Symphony, a German composer known for his iconic works such as "Moonlight Sonata" and "Für Elise." Answer: Ludwig van Beethoven`

The question, from a subjective point of view, could be a valid quizbowl question. Compared to GPT-2, the performance increased visibly. Also, the first two sentences are related semantically. However, the model sometimes also generates shorter questions with incomplete sentences:

`Question: This book includes the line "All animals are equal, but some animals are more equal than others." The pigs Napoleon and Snowball vie for control of Manor Farm in, for 10 points, what novel by George Orwell? Answer: Animal Farm.`

When GPT-4 [6] is simply prompted to provide quizbowl questions [4], we get a result such as the following:
`It's not dopamine, but this molecule's reuptake is inhibited by Prozac. A deficiency of this molecule is thought to contribute to depression. For 10 points, name this neurotransmitter, commonly referred to as the "feel-good" hormone. Answer: Serotonin`

Inspecting the results from GPT-4, without fine-tuning or being few-shot prompted, the questions seem to be properly constructed in the quizbowl format most of the time, despite the fact that there were some obvious logical mistakes such as giving the answer in one of the clue sentences. Also, the question is quite short.

Ultimately, all of these approaches are great baselines, and baselines that our model should be able to outperform. Following is a spreadsheet of more sample baseline questions.

Baseline Results

4 Methods and Architecture

Currently, we have 3 methods of generating questions: Few-Shot Prompted GPT4, PyraModel, and a DPR-based method. Currently, only the first two create intricate high-level questions and are being evaluated by humans on the following website. There are over 330 questions on this website if you wish to see specific examples of our questions.

4.1 Few-Shot Prompted GPT4

This is essentially an improved method to our baseline approaches, which incorporates a few examples of previously written HS-level QuizBowl questions to fine-tune the performance of GPT4 to specialize in generating QB questions. We also explain the concept of pyramidality to it. It utilizes its own knowledge of the topic as well as text-generation techniques to create its best attempt at a QB question without directly being told what content to include within clues. It is five-shot prompted on five previous good questions.

4.2 PyraModel

Brief Overview: We begin with a topic (which for simplicity we take to be a Wikipedia page title). We then use our "black-box" PyraModel to decide which aspects of that topic are unique, interesting, and of the appropriate difficulty to ask about. Then we compose those clues together as Wikipedia sentences in a specific order and pass them to a few-shot prompted GPT4 to create a single grammatically coherent and flowing question.

For our novel PyraModel approach to solving a novel task (generating good QuizBowl questions), we fine-tune a pre-trained Llama2 model for a regression task (the transformers API recognizes this as a Sequence Classification task with only a singular "continuous float" label). We train the model on every clue (which maps to a sentence) of all ? high-school level questions in the 2021 QANTA train dataset. The text is the clue text itself and the regression label is the positional value of that clue in the question in the range (0, 1]. The latter value is determined by

$$label = \frac{clue\_num}{num\_clues} \tag{1}$$

Extra multimodal features are also appended for each clue. There are 2 extra categorical features: the topic name and the tournament name that the question appears in. There is a numerical feature of length 50 attached as well: the similarity of that clue's text to 50 chunks of that topic's corresponding Wikipedia article, where the article is split in such a way that each of the 50 chunks is of similar token length while retaining whole sentences for each. This is determined by taking the BERT embedding of the clue text as well as each segment and then calculating the cosine-similarity scores between these embeddings.

After the training process is complete, we move on to the first step of our two-step QB question generation pipeline: the retriever.

4.2.1 Step 1: Retriever. In this step, we simply want to gather the information we wish to include within each clue of the question for the chosen topic. Thus, we run the model on every sentence in the Wikipedia article to figure out "where" each sentence belongs and how difficult/obscure those facts are. Then, for questions of however many clues we want to generate (say n clues), we assign a number to each sentence ranging from 1 to n depending on which of the n buckets the predicted clue_pos of that sentence falls within. Thus, we can randomly sample first clues, second clues, ..., and last clues (in terms of facts and semantic components) for a new question about that topic, and eventually merge these together to build a question from this. You may wonder, isn't it a major issue that we are running the model on every sentence in the article? By assigning a bucket to each sentence and randomly sampling from those, we don't know if the chosen sentence is unrelated filler-text or interesting as a clue. For the first issue, we originally considered running some form of Named Entity Recognition with a threshold of minimum entities per sentence; however, after some testing, we noticed that almost all Wikipedia sentences are quite densely packed with information and details, so there is no

shortage of unique and true content in each clue; our generated questions reflect this. The latter is quite a subjective metric that is difficult, and we decided to implement a classifier to prune clues/sentences that contain "uninteresting" content.

Interestingness Classifier Through the interestingness classifier, we focus on pruning retrieved information that is not deemed interesting or related as a clue, and utilizing PyraModel to re-sample another fact. This classifier is trained on a 50/50 distribution of early QB question clues (every clue except last) for a topic (considered as interesting/salient) and Simple Wikipedia sentences of that topic (considered as uninteresting/boring). For the uninteresting train set, we use the corresponding Wikipedia sentences of each Simple Wikipedia sentence, determined through text similarity; this is done so that the model doesn't hyperfocus on the simpler linguistic differences of Simple Wikipedia as opposed to the semantic differences. It is also important to note that usually, the content of Simple Wikipedia corresponds to a subset of normal Wikipedia. Currently, we are sampling from all sentences correlated with a clue #; however, once this classifier is done being trained (in the training process at the current moment), we will only consider a certain subset of those sentences when sampling. This is our retrieval portion and arguably the more difficult and important part: determining what content to include in each clue.

4.2.2 Step 2: Generator.

The second step is the generator, which simply uses GPT4 to structure all the content together in the same order as the provided clues and in the format of a QB question; it is also imperative to ensure the question flows smoothly. We two-shot prompt GPT4 for this. Following are our prompts:

```
"Write a difficult ? sentence quizbowl question with answer ? that includes information from the following attached sentences in order. Make sure that the order of information in the question remains the same as the input. Put the final generated question in the format of a paragraph. Do not include the word ? or similar words in this question, as we don't want to reveal the answer in the question itself. Don't add any information/details not provided in the input. The input sentences are separated as elements of this list of ? strings: ?."
```

```
"Rewrite sentences of the following paragraph that include ? or similar words to get rid of those words so the answer to the quizbowl question is not obvious. Also, rewrite the last sentence of the paragraph to start with "For 10 points, name this ...". Do NOT rewrite any other sentences or information. Here is the paragraph: ?"
```

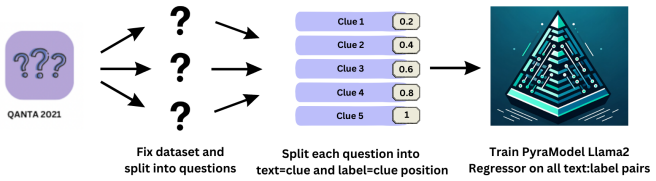

Figure 2. Training of PyraModel

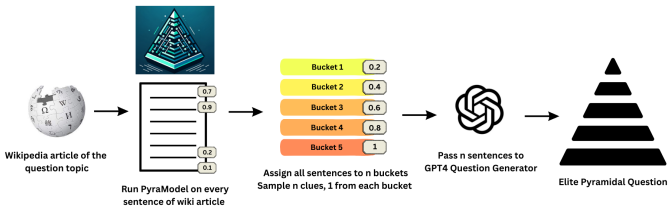

Figure 3. Question Generation Scheme of PyraModel

5 Automatic Computer Evaluation

Questions are being evaluated by Professional writers and players on the website, but we are also evaluating the model on the QANTA test set to see performance in terms of measuring difficulty of clues. The plots below represent the difference between the Predicted clue position (difficulty) and the Ground Truth clue position of clues in the test set questions. Overall, as we can see in the violin plot, the first and last clues seem to generally be quite perfect, and middling clues have an upward trend between the predicted position and GT position but are much looser in terms of precision and encapsulate a wider range of possibilities. This makes sense, as QuizBowl questions vary vastly and are designed in such a way that they don't all follow a directly linear difficulty curve, but rather, just a general trend. This subjectivity and unique "randomness" between questions is what keeps the game fresh and interesting for players.

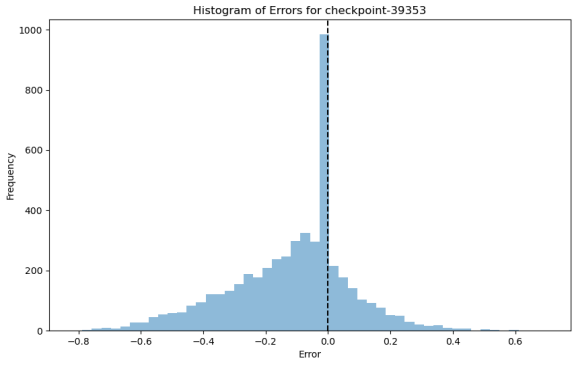

Figure 4. Histogram of Regressor QANTA Test Set Errors

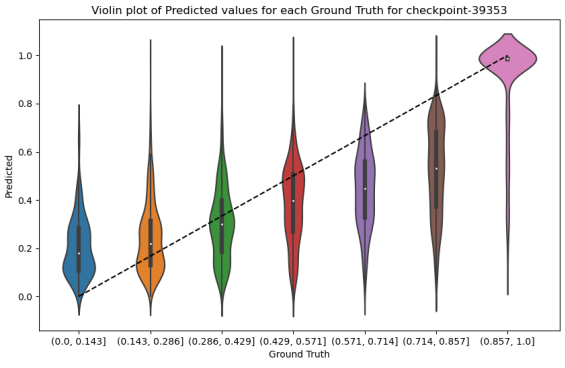

Figure 5. Violin Plot of Regressor QANTA Test Set Errors

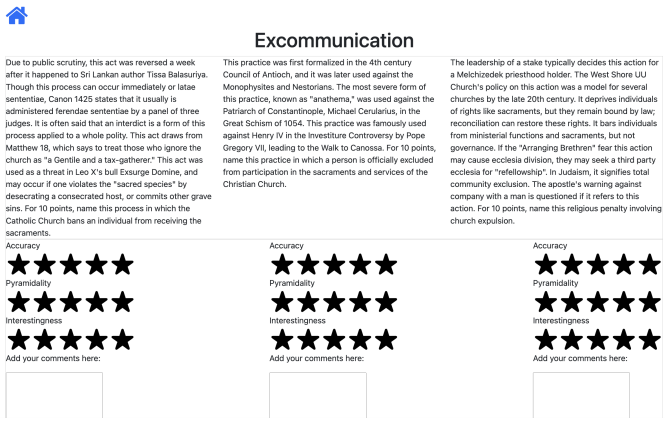

Figure 6. Layout of the Human-Feedback Platform

Here is a spreadsheet containing some specific textual examples of the most and least egregious errors (furthest away and closest together). Definitely a lot of randomness, but this seems inherent to the task at hand.

6 Human Evaluation

In the context of text generation, it is often subjective and intrinsically difficult to assess generated text. This applies to our problem as well; it is hard to evaluate generated questions and to define metrics to evaluate and optimize the model. Since the feedback mechanisms are unfeasible to maintain and time-consuming for humans, there are proposed metrics such as BLEU[7] and FActScore [5]to assess generated text automatically. However, real feedback from human evaluators is best to capture human preference. This is where professional QuizBowl writers and players come into play. In the website we deployed, users sign in with a unique username to rate questions about different topics. There are 110 topics and 330 total questions, from 3 methods. One of these is real human-written questions found in previous tournaments, and the other 2 are our own computational methods (Few-Shot Prompted GPT4 and the PyraModel Regressor). For each question, the users rate it in comparison to the others on the 3 metrics shown (Accuracy, Pyramidality, and Interestingness). There is also a textbox to Write any specifics about the question and evaluation.

7 Enhancing the Questions with RLHF

The feedback can then be used to enhance the model with reinforcement learning methods. In this specific case, we will be using something called Reinforcement Learning from Human Feedback (RLHF). The high-level idea of reinforcement learning in this context is to update the policy of the model according to the output of a reward model so that the rewards will be optimized and the outcome will be closer to what is expected by humans. The adoption of reinforcement learning for our problem is as follows, we associate our method of question generation with three additional entities: a feedback collection platform, a reward model, and an additional fine-tuning step.

Feedback Platform: We deploy a website to gather feedback from professional QuizBowl question writers and players, on 3 sets of questions on a random QuizBowl topic. On every feedback attempt, the evaluator will be presented with one sample from each class of questions. The evaluator needs to assign scores out of 5 for pyramidality, accuracy, and interestingness. A layout of the evaluation page is presented earlier. As the names imply, pyramidality is a measure of whether the question consists of multiple clues arranged in descending order of difficulty, that is, with the hardest information first and the easiest at the end; accuracy measures whether the question is grammatically and factually correct; interestingness is the metric measuring whether the question consists of unique facts. For this project, we will focus only on accuracy and pyramidality since our primary goal is to create factually correct and pyramidal questions compatible with the QuizBowl format. We leave the utilization of interestingness to a later study. The metrics that we use, other than accuracy, are highly subjective and depend on the perspective of the evaluator. Thus we select our evaluators to be experienced quiz bowl players and question creators. For our purpose of obtaining a reward model, a better scoring system would be to ask for scores for individual clues separately, however, this would be inconvenient and time-consuming for the evaluators. Thus, we later need to adapt these scores to our reward model accordingly.

Reward Model: In a reinforcement learning system, the reward model is essential since it directs the learning loop. In the context of our method, the reward model needs to reflect the quality of the question, more specifically the positional value attached to a Wikipedia sentence, generated

by the model. For our implementation, PyraModel, the reward model is another pre-trained language model that is fine-tuned for another regression task, which is to output quality scores. We train the model on the clue text and positional difficulty pairs along with user-provided feedback as two additional features. In our case, there are two types of input: accuracy and pyramidality scores, both of them being out of 5. For feedback provided for a generated question, each sentence (clue) will have the same accuracy and pyramidality score, which assumes that each part of the question contributes equally to its overall quality. We scale both of the scores into the range [0,1] and train the model with an individual clue, accuracy, and pyramidality score. At the end of this process what we expect is to obtain a model which will take the clue (Wikipedia sentence) and predicted positional (difficulty) value for that as inputs, and will output a scalar indicating the quality of the positional value. Following that, we optimize the original language model in relation to the reward model using reinforcement learning.

Fine-Tuning with respect to Reward Model: In this step, we fine-tune a copy of the initial model, PyraModel, with a policy-gradient reinforcement learning algorithm; Proximal Policy Optimization [12]. PPO is used for this task because it is efficient and effective at handling the complex policy space involved in natural language processing tasks. The PPO algorithm fine-tunes the parameters of PyraModel in such a way that the reward predicted by our reward model is maximized. To continue to enhance the overall quality of the generated questions in terms of how well they match with the human-provided feedback on accuracy and pyramidality, the model continues its iterative process of fine-tuning by creating new questions, obtaining updated rewards, and modifying its parameters. By doing this, PyraModel improves its ability to generate better quality questions that are adjusted to the expected difficulty increase in pyramid-style questioning.

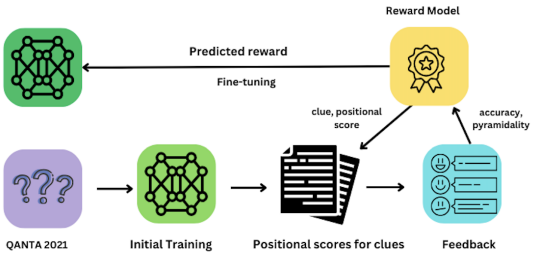

Figure 7. PyraModel's RLHF Pipeline

RLHF for Few-Shot Prompted GPT4: For Few-Shot Prompted GPT4, RLHF is easy, as we can simply change the prompts depending on the feedback we receive, and then reevaluate and repeat the process until we receive good results. We have briefly tied this and received good results, as this is essentially few-shot learning in the context of LLMs

Important Note: We unfortunately did not get much human feedback by the time this scholarly paper is going to be submitted, but the pipeline is in development and we are awaiting human responses and evaluation.

8 Wrapping Up

8.1 Improvements and Future Work

8.1.1 Interknitted Adjacent Clues. Throughout our study and generation of QuizBowl questions, we only considered clues as singular entities independent of one another, hence our direct mapping of clues to sentences. However, in reality, clues may extend across multiple sentences, each elaborating on top of the other. Although we are still able to keep a smooth flow between sentences, each adjacent sentence is a completely different clue hinting at the same answer. In the case of semantic similarity between clues, it may be somewhat trivial to implement by scanning the corresponding Wikipedia article for similarity through our own coded methods. However, sometimes we have closely related clues that are not close in terms of word semantic similarity, such as a clue about a country's GDP and one about its population; this case is quite complex and seems like a difficult task for a computer.

8.1.2 Uniquely Identifying Clues. In most QB questions created in 2014 and later, clues generally uniquely identify the answer. One possible way to evaluate this is by running a QA system on the clue to see if closely related topics are confused and are predicted at nearly the same frequency. For example, in our current framework, a clue sampled from facts in the Wikipedia article about Yellow Fever could also be true for the Bubonic Plague. We're not quite sure what the best approach is to implement this, but currently considering a form of contrastive learning, where we look at facts about similar topics that are not true about the current topic. By the time this scholarly paper is submitted, this will likely not be implemented, but we will start working on specificity soon.

8.2 Conclusion

In conclusion, our PyraModel and Few-Shot-Prompted GPT4 approaches far surpassed the baseline GPT2 and GPT3.5 methods. From what we've seen, PyraModel did in fact surpass the GPT4 approach in terms of both pyramidality and accuracy, and we hope that our work with conditioning Large Language Models on factual text will prove useful not only for the domain of QuizBowl but also the future

of LLMs. Although the quality of questions is not as good as human-written ones, this makes sense as those require a lot of knowledge, effort, and money, and our questions can be efficiently and cheaply computationally generated. It is also quite important to note the sheer importance of human intervention in NLP and the current state of LLMs, to combat hallucinations and ensure relevancy of text. We hope that once we get more real human-feedback, both approaches/models will perform even better.

References

[1] Emily M. Bender, Timnit Gebru, Angelina McMillan-Major, and Shmargaret Shmitchell. On the dangers of stochastic parrots: Can language models be too big?, 2021.

[2] Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners, 2020.

[3] J. Carlson. Writing quizbowl questions with gpt-2. Medium, 2020. Available at: https://medium.com/@jaimie.m.m.carlson/writing-quizbowl-questions-with-gpt-2-1c3839a3f39c.

[4] D. M. Levinson. Transportist: A set of quizbowl practice questions, by gpt-4. https://www.transportist.net/p/a-set-of-quizbowl-practice-questions, 2023.

[5] Sewon Min, Kalpesh Krishna, Xinxi Lyu, Mike Lewis, Wen tau Yih, Pang Wei Koh, Mohit Iyyer, Luke Zettlemoyer, and Hannaneh Hajishirzi. Factscore: Fine-grained atomic evaluation of factual precision in long form text generation, 2023.

[6] R OpenAI. Gpt-4 technical report. *arXiv*, pages 2303–08774, 2023.

[7] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. Bleu: A method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, ACL '02, page 311–318, USA, 2002. Association for Computational Linguistics.

[8] Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. 2019.

[9] Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. Squad: 100,000+ questions for machine comprehension of text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2383–2392, Austin, Texas, 2016. Association for Computational Linguistics.

[10] Pedro Rodriguez, Shi Feng, Mohit Iyyer, He He, and Jordan Boyd-Graber. Quizbowl: The case for incremental question answering, 2021.

[11] Yingqi Qu Wayne Xin Zhao Jing Liu Hao Tian Hua Wu Ji-Rong Wen Haifeng Wang Ruiyang Ren, Yuhao Wang. Investigating the factual knowledge boundary of large language models with retrieval augmentation, 2023.

[12] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms, 2017.

[13] Iulian Vlad Serban, Ryan Lowe, Laurent Charlin, and Joelle Pineau. Generative deep neural networks for dialogue: A short review, 2016.

[14] Johannes Welbl, Pontus Stenetorp, and Sebastian Riedel. Constructing datasets for multi-hop reading comprehension across documents, 2018.

[15] Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William W. Cohen, Ruslan Salakhutdinov, and Christopher D. Manning. Hotpotqa: A dataset for diverse, explainable multi-hop question answering, 2018.