Cheater's Bowl: Human vs. Computer Search Strategies for Open-Domain Question Answering

Wanrong He
Tsinghua University*
hewanrong8@gmail.com

Andrew Mao
University of Maryland
amao1@terpmail.umd.edu

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

Abstract

For humans and computers, the first step in answering an open-domain question is retrieving a set of relevant documents from a large corpus. However, the strategies that computers use fundamentally differ from those of humans. To better understand these differences, we design a gamified interface for data collection—Cheater's Bowl—where a human answers complex questions with access to both traditional and modern search tools. We collect a dataset of human search sessions, analyze human search strategies, and compare them to state-of-the-art multi-hop QA models. Humans query logically, apply dynamic search chains, and use world knowledge to boost searching. We demonstrate how human queries can improve the accuracy of existing systems and propose improving the future design of QA models.

1 The Joy of Search: Only for Humans?

A grand goal of artificial intelligence research is to design agents that can search for information to answer complex questions. Modern day question answering (QA) models have the ability to issue text-based queries to a search engine (Qi et al., 2019, 2021; Xiong et al., 2021; Zhao et al., 2021; Adolphs et al., 2022a; Nakano et al., 2021) and use multiple iterations of querying and reading to search for an answer. However, there is still a performance gap between machines and humans.

Dan Russell describes humans with virtuosic search abilities in his book *The Joy of Search* (Russell, 2019a), and describes search strategies that: use world knowledge; use parallel search chains, abandon futile threads; and use multiple sources and languages. While we can all admire Dan Russell's search skills, it does not answer the question: how different are computers' searches from those of humans?

This paper attempts to answer this question with a comparison of human and computer search strategies. We create "Cheater's Bowl", an interface that gamifies answering questions, with the addition of tools such as a traditional search engine, a neural search engine, and modern QA models. We collect a dataset of human search sessions while using our interface to answer complex open-domain multi-hop questions (Section 3) from Quizbowl (Rodriguez et al., 2021, QB) and HotpotQA (Yang et al., 2018). We analyze the differences between human and computer search strategies and detail where current models fall short (Section 4). Substituting queries generated by models with human queries significantly improves model accuracy. We propose design suggestions for future query-driven QA models, such as creating retriever-aware queries and validating answers. Our dataset can serve as the foundation for training them (Section 5).

Our main contributions are:

- We create an interface for answering questions with access to search tools.
- We collect a dataset of human search sessions on questions from Quizbowl and HotpotQA.
- We compare human and computer strategies for QA: humans apply dynamic search chains, use world knowledge, and reason logically. We propose these as potential directions for query-driven QA models.

2 How Humans and Computers Search

To compare how humans and computers form queries to answer questions, we first need to have a level playing field and set up our vocabulary. Sometimes, we will need to speak abstractly about who is trying to answer the question without distinguishing between the human and the computer. In these cases, we refer to them as an "agent", which can be either the human or the computer. We assume that the agents do not know the answers directly and

*Work completed at University of Maryland.

Findings of the Association for Computational Linguistics: EMNLP 2022, pages 3627 - 3639

that they create text-based queries to find the answer (we discuss the alternatives, closed book QA, directly forming dense queries and other computer systems, in Section 6).

We assume that agents, given an initial question, form a text query q_0. The i^{th} query q_i retrieves a set of documents $\mathcal{D}_{i+1} = \{d_1, \ldots, d_{|\mathcal{D}_{i+1}|}\}$ from a large corpus of documents $\mathcal{D}$—in our case all the paragraphs in Wikipedia pages. The retrieved documents provide additional information, allowing the agent to answer the question or compose a new query q_{i+1}. The set of documents $\mathcal{E}_i \subseteq \mathcal{D}_i$ provide information—evidence—for answering the question with answer a or composing subsequent queries $\{q_j | j > i\}$. It is possible that not all retrieved documents are read—$\mathcal{E}_i \neq \mathcal{D}_i$—since not all of the retrieved documents are relevant, and an agent might only read a few of them. This process repeats until the agent answers the question. We represent the iterative question-answering process as action path: $A = (q_0, \mathcal{E}_1, q_1, \mathcal{E}_2, q_2, \cdots, \mathcal{E}_k, a)$.

2.1 Human Queries

How humans form queries when they search for an answer depends on many factors, as summarized by Allen (1991): the experience of the user searching for information, how much the user knows about the topic, and whether they are finding completely new information or navigating to a specific information source they have seen before. Beyond the intrinsic knowledge of particular users, users often have particular strategies they favor. For example, users may copy/paste information into a document, keep multiple tabs open, or always turn to a particular source of information first (Aula et al., 2005).

2.2 Computer Systems

Thanks to the recent development of machine learning and natural language understanding, computer systems can answer open-domain questions by generating text-based queries. The GOLDEN retriever (Qi et al., 2019) generates a query q_k at reasoning step k by selecting a substring from the current reasoning path R_k, which is the concatenation of the question Q and previously selected retrieval results at each reasoning step: $R_k = (Q, d_1, d_2, \cdots, d_k)$, $R_0 = (Q)$ (for questions with $n \geq 1$ clues/sentences, we use their concatenation as the full question $Q = (Q_0, Q_1, \cdots, Q_{n-1})$). GOLDEN retriever then selects a single document d_{k+1} from the set of documents $\mathcal{D}_{k+1}$ retrieved by q_k, appends d_{k+1} to the current reasoning path and forms an updated reasoning path R_{k+1}. IRRR (Qi et al., 2021) further advances GOLDEN retriever by allowing queries to be any subsequence of the reasoning path, though less flexible than human queries. At each step, these systems only select a document as evidence for further actions: $\mathcal{E}_i = \{d_i\}$. Thus the action path becomes $A = (q_0, \mathcal{E}_1, q_1, \mathcal{E}_2, q_2, \cdots, \mathcal{E}_k, a) = (q_0, \{d_1\}, q_1, \{d_2\}, q_2, \cdots, \{d_k\}, a)$.

3 Cheater's Bowl: Gamified Data Collection For Human Searches

This section discusses the gamified data collection process via an example.

3.1 Motivation

High-stakes trivia competitions test who knows more about a particular topic. However, it has occasionally been plagued by cheater scandals from Charles van Doren in the 1950s (Tedlow, 1976) to Andy Watkins in the 2010s (Trotter, 2013). These scandals are not particularly relevant to computational linguistics, but the move to online trivia competitions during the Corona pandemic brought a new form of cheating to the fore: people would see a trivia question and quickly try to use a search engine to find the answer.

Some of the online discussions around online cheating revealed that some people actually enjoyed doing these quick dives for information. Thus, a goal of this paper is to see if we could (1) sublimate these urges into something more wholesome, (2) gather useful data to understand human expert search, and—a less scientific question—(3) see who is the best at cheating in trivia competitions. To answer these questions, we create a gamified interface (Figure 1)—which we call Cheater's Bowl—to help players find answers.

Because players come from the trivia playing community, they know substantially more about the topics than, say, crowdworkers. This puts them closer to the "expert" category discussed by Allen (1991). We use Quizbowl questions (Boyd-Graber et al., 2012, QB), where each question is a sequence of clues with the same answer of decreasing difficulty (as decided by a human editor). We also include questions from HotpotQA (Yang et al., 2018), a popular dataset for multi-hop question answering. We filter the questions in two ways to ensure that both humans and computers are challenged. First, we discard all but the two hardest clues, which

should be difficult for most humans (even our experienced player base). Second, we try to answer all of these questions with current state-of-the-art BERT-based model on the data (Rodriguez et al., 2021) with a single hop. If the model can answer the question with any number of clues, we exclude it from the questions set used in the data collection.

3.2 Game Interface

The player is presented with a question, initially with only one clue. To start searching, the players have the option of typing their own queries in the search box, or clicking on a model-suggested query (from IRRR or GOLDEN). The search engine returns results from two different retrievers: BM25, a sparse index based on lexical similarity; and Dense Passage Retrieval (Karpukhin et al., 2020, DPR), which uses dense vector embeddings of passages. Both retrievers index and return paragraphs from Wikipedia pages. We use ElasticSearch (Gormley and Tong, 2015) to implement BM25, and for DPR, we directly use a pretrained model.

Both retrievers return the top passages by cosine similarity. Players can click on the Wikipedia page titles of the passages; the full Wikipedia page is then shown in the main document display with the passage highlighted.

The popup tooltip provides shortcuts to directly query the search engine from highlighted text, record it as evidence, or submit it as an answer. Players are encouraged to highlight and record text as evidence if it is helpful for them to find the answer. Even if a player does not record any evidence, the paragraphs a player reads are automatically recorded as evidence.

If the player finds the question difficult to answer, they are free to skip the question or ask the system to reveal another clue.[2]

Human-computer collaboration. In addition to the queries from GOLDEN and IRRR, players also see IRRR's answers. Players can directly answer the question with suggested answers (but are encouraged to find evidence to back it up).

Scoring system. Our goal is to create an interface that is both fun and useful for collecting relevant information. Players are rewarded for having the highest score, and they earn points by: (1) answering more questions, as each question adds to their score; (2) answering questions correctly (100 points for each correct answer); (3) answering quickly, as the possible points decrease with a timer (four minutes for QB questions, three for HotpotQA); (4) answering with fewer clues, as it makes the question easier (each clue removes ten points); (5) recording more evidence. Each piece of recorded evidence is awarded ten points.

3.3 The Player Community

We recruit thirty-one players from the trivia community who played the game over the course of the week. The top player answered 895 questions, and thirteen players answered at least forty questions. After filtering out empty answers and repeated submissions of the same player on the same question, we collect 2545 questions-answer pairs from QB of which 1428 were correctly answered (56%), as well as 315 pairs from HotpotQA, of which 225 were correctly answered (71.43%).

3.4 A Question Answering Example

To see how a player might answer the question with our interface, we present a question-answering example with corresponding player actions (Figure 2). Answering this question requires figuring out who the main speaker was (Prem Rawat) and then figuring out his nationality to get to the final answer, India. The player answers the question with two hops: first to "Millennium '73" and then to "Prem Rawat", finally uses commonsense reasoning to answer "India". Player actions and seen paragraphs are automatically recorded through the process.

4 Human vs. Computer Search Strategies

This section compares and contrasts how computers and humans search for information.

4.1 Strategies in Common

Agents search Wikipedia using text queries, process results, and give an answer. Both humans and computers often create queries from the question: 83.05% of human queries have at least one word from the question, while 84.61% of GOLDEN queries and 99.75% of IRRR queries do. Both use terms from the evidence they find to create new queries: 14.47% of human queries have at least one word from retrieved evidence, while 19.13% of GOLDEN and 28.30% of IRRR queries do. Both reformulate their queries based on evidence to uncover new information (Xiong et al., 2021).

[1]The figure is a screenshot to illustrate the interface during player training. The question is not a part of the experiment data.

[2]This only applies to QB questions.

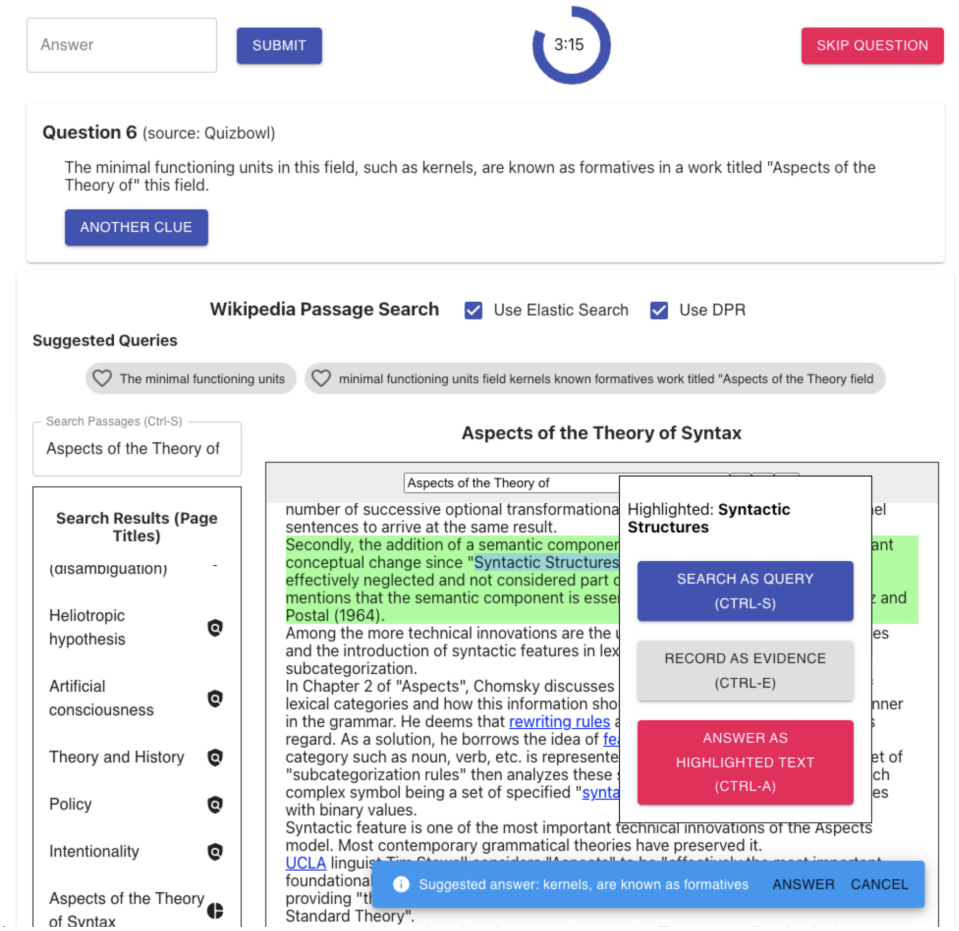

Figure 1: User interface for Cheater's Bowl, an interface to collect user traces as they try to answer difficult questions. Players see a question[1] (top), can search for information (left), view information (center), and give their answer (top) with associated evidence (right).

4.2 Where Strategies Diverge

Humans use fewer but more effective keywords. The most salient difference between human and computer queries is that human queries are shorter. Human queries contain 2.67 words on average (standard deviation of 2.46); while GOLDEN retriever's contain 7.03±6.84 words, and IRRR's have 12.76±5.64. Human queries focus on proper nouns and short phrases (Figure 3). Humans tend to select the most specialized term—e.g., the entity most likely to have a comprehensive Wikipedia page—which requires world knowledge (Table 1). In contrast to humans' desire for precision, models prefer recall with as many keywords as possible, hoping that it retrieves something useful for the next hop.

Humans use world knowledge to narrow search results. Unlike computers, humans sometimes use words that are not in the question or in evidence: 16.30% of queries have terms in neither evidence nor question text (compared to 0% for both computer methods). In the first example in Table 1, the player's first query is derived from the question but adds "auxiliary", recognizing that "treating" a compound makes it an auxiliary in the reaction. Players also reported in the feedback survey that adding a subject category (for example, adding "chemist" when querying a person in chemical-related questions) can be useful for restricting the search results. Although there are cases when players directly query terms closely related to the answer, in most cases, people use common sense to narrow the search scope or use domain-specific knowledge they have learned from previous searches. These patterns could be potentially learned by QA models, as we discuss further in Section 5.

Dynamic query refinement and abandonment. Although both humans and computers reformulate queries as a search strategy, how humans reform

Question: "A 15-year-old religious leader originally from this country spoke at a highly anticipated event at which it was predicted that the Astrodome would levitate; that event was Millennium '73". **Answer**: "India".
(1) Query q_0 ="Millennium '73" (Substring of question)
(2) Select and read the Wikipedia page: "Millennium '73". Manually record evidence d_1 =" It featured Prem Rawat, then known as Guru Maharaj Ji, a 15-year-old guru and the leader of a fast-growing new religious movement."
(3) Query q_1 ="Prem Rawat" (Substring from evidence d_1)
(4) Select and read the Wikipedia page: "Prem Rawat". Manually record evidence d_2 ="Prem Pal Singh Rawat is the youngest son of Hans Ram Singh Rawat, an Indian guru."
(5) Answer a ="India" (Derived from evidence d_2)

Figure 2: An example of player actions for question answering with action path $A = (q_0, \mathcal{E}_1, q_1, \mathcal{E}_2, a)$, where $\mathcal{E}_1 = \{d_1\}$ and $\mathcal{E}_2 = \{d_2\}$. The player uses substrings from the question and evidence as queries, and derives the final answer from evidence. We highlight the source of actions in blue. Our goal is to use these interactions to better understand computers' question answering.

Question and answer	First query		
	Player	IRRR	GOLDEN retriever
Q: Evans et al. developed bisoxazoline complexes of this element to catalyze enantioselective Diels-Alder reactions. A: Copper	Evans **auxiliary**	Evans **et al. developed bisoxazoline complexes element catalyze enantioselective Diels-Alder reactions**	Evans **et al.**
Q: This quantity's name is used to describe situations in which there exists a frame of reference such that two given events could have happened at the same location. A: time	frame of reference **same** location	**quantity's name used describe situations exists** frame reference **two given events could happened** location	**quantity's name is used to describe situations**
Q: Discovered in 1886 by Clemens Winkler, this element is used in glass in infrared optical devices, its oxide has been used in medicine, and its dioxide is used to produce glass with a high index of refraction. A: Germanium	Clemens Winkler	**Discovered 1886 Clemens** Winkler element **used glass infrared optical devices oxide used dioxide used glass high index refraction**	**Discovered in 1886 by** Clemens Winkler
Q: In ruling on these documents, the Court held that the "heavy presumption" against prior restraint was not overcome. A: Pentagon Papers	heavy presumption prior restraint	**ruling documents Court held** "heavy presumption" **against prior** restraint overcome	**ruling on these documents, the Court**
Q: One of this director's films introduced the cheery song "High Hopes," while another describes the presidential campaign of Grant Matthews. A: Frank Capra	high hopes song	**One director's films introduced cheery** song "High Hopes " **describes presidential campaign Grant Matthews**	**director's films introduced the cheery** song "High Hopes,"

Table 1: The first query for each question from different agents. Computers' words that are distinct from humans' are in **bold**. Human queries contain fewer keywords and focus more on precision, while computer queries focus more on recall.

their queries is more advanced. Not all retrieved documents help lead to the answer: some are irrelevant, and some are even misleading. In cases when human agents have not found any helpful information from the documents $\mathcal{D}_i$ retrieved by query q_i or when they are confused and unsure, the human agent does not need to use a document from $\mathcal{D}_{i+1}$ for making new queries. If that happens, they ignore the useless evidence, i.e. $\mathcal{E}_{i+1} = \emptyset$. Instead, they write a new query q_{i+1} by adding more constraint words and deleting distracting terms from q_i to restrict the search scope or abandon q_i and write a completely new query. Russell (2019b) describes querying "stoplight parrotfish sand" to uncover the relationship between parrotfish and geology, however, the results are too mixed to be useful. He then modifies the query to "parrotfish sand", which yields good results.

However, for GOLDEN retriever and IRRR, even when irrelevant documents are retrieved from a bad query q_i, the model is compelled to select some $d_{i+1} \in \mathcal{D}_{i+1}$ as evidence, append to the reasoning path, and generate subsequent queries accordingly. As an example, to answer the question

> He lost the presidential election in 1930, which was not good enough for him as later that year he seized power at the head of an army-backed coup. (Answer: Getúlio Vargas (a Brazilian president))

IRRR queries "lost presidential election 1930 year seized power head army backed coup" but an article

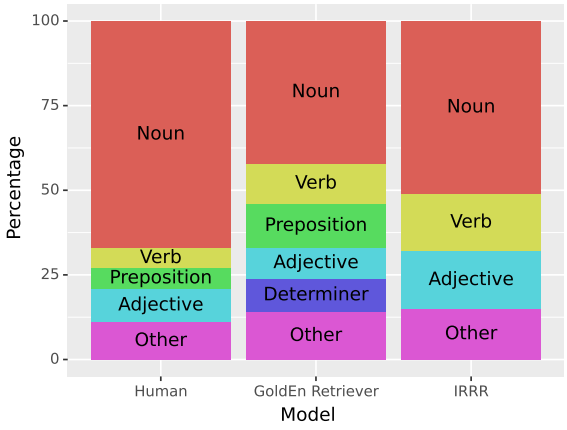

Figure 3: Proportion of different part-of-speech tags used in queries detected by the the Natural Language Toolkit (Bird et al., 2009, NLTK). Humans focus more on nouns, eschewing determiners for the sake of brevity. On this basis, IRRR is the more "human" of the two computer agents.

about Brazil is not in the returned results. IRRR then appends a paragraph from the irrelevant page about the Nigerian president "Olusegun Obasanjo" to the reasoning path, leading to the next query "lost presidential election 1930 later year seized power head army backed coup Olusegun Obasanjo" which prevents finding a relevant Brazilian page.

Multiple search chains. A search chain is searches $(q_s, q_{s+1}, q_{s+2}, \cdots, q_t)$. New searches depend on old ones, either because a subsequent search q_{i+1} refines a previous search—q_i or q_{i+1}, for example—integrates evidence $\mathcal{E}_{i+1}$ retrieved from q_i. A search chain breaks when q_i is abandoned and q_{i+1} is unrelated to previous evidence or queries. While existing computer agents can only use a single search chain, human agents use multiple search chains, either pre-planned parallel search chains that focus on different perspectives of the question, or starting a new one if previous chains fail to lead to the answer. When answering the question

> This modern-day country was once ruled by renegade Janissaries known as dahije, who massacred this country's elite, known as knez, in 1804. (Answer: "Serbia")

the player first makes a query about the mentioned title "knez", and next queries "Knyaz", which is a substring of the evidence retrieved by the first query. However, these queries fail to retrieve useful results since "knez" and "Knyaz" are common Slavic titles. The player then abandons this search chain and starts a new one with the query "dahije", which allows the player to retrieve the Wikipedia page "Dahije" that includes the answer "Serbia".

Swapping Engines. The *Joy of Search* is replete with searches over different sources: Google, Google Scholar, Google Earth, etc. While we only give players access to Wikipedia, we allow players to switch between ElasticSearch and DPR. In contrast to multi-hop systems which typically use trained, dense retrievers, players prefer ElasticSearch (87% of queries) over DPR. Some of this is probably familiarity: most public-facing search engines (including Wikipedia) are term-based retrievers. In the post-task survey, players prefer ElasticSearch because it is most useful when looking for an exact Wikipedia page—the specific Wikipedia page always ranked top among search results. It is also helpful for checking answers: they often query an answer candidate, which helps boost their answer accuracy. ElasticSearch—given its predictability—is better for this specific strategy.

Beyond a Bag of Words. However, this is not always the case; when humans do use DPR, they adapt their query styles for better retrieval. Some players report that they could find their desired results with natural language queries when using DPR. Those queries usually come from longer sequences in the question or evidence. For example, when answering the question

> Mathilda Loisel goes into debt to replace paste replicas of these gemstones, one of which is "As Big as the Ritz" in an F. Scott Fitzgerald short story. (Answer: "Diamond")

the player queries "As Big as the Ritz" in an F. Scott Fitzgerald short story." with DPR, which retrieves the Wikipedia page "The Diamond as Big as the Ritz" with the answer.

Players also report searching Google with natural language queries when finding answers to open-ended questions with various options, e.g., "How often should I wash my car?". In these scenarios, humans may search for relatively vague queries and synthesize an answer from multiple retrieval results. WebGPT (Nakano et al., 2021) explores a similar setting by training GPT-3 (Brown et al., 2020) to search queries in natural language, aggregate information from multiple web pages and answer open-ended questions. Due to the limitation of Cheater's Bowl where most QB answers are Wikipedia titles (Rodriguez et al., 2021), agents do not need this more flexible setup.

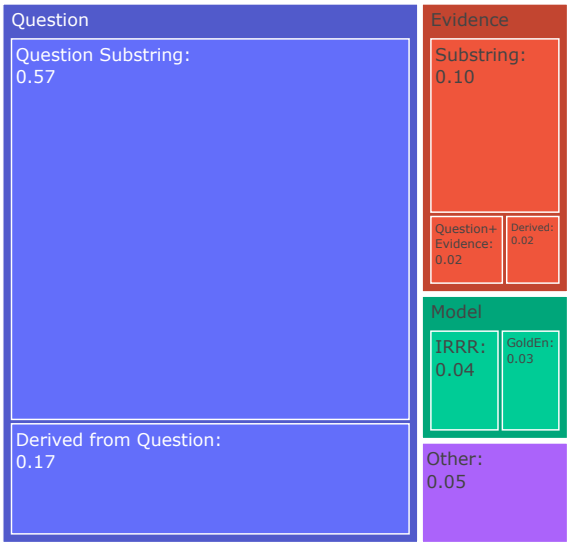

Figure 4: Users need to find the answer to a question but have several sources that might inspire their queries: the original question, evidence, or models. This Treemap shows the source for their questions (area corresponds to frequency). Only a small proportion of queries are from QA models' suggestions.

5 Existing Models and Future Design

Although we present queries suggested by state-of-the-art multi-hop QA models to players, players would rather write their own queries (Figure 4). Most players understand why QA models query the way they do (Figure 5) and agree that queries retrieve helpful results, but players question the utility. This is an intrinsic difference between humans and models: human queries strive for a "direct hit" with two to three search results, as (Jansen et al., 2000) find that humans only access results on the first page, typically the first few results. In contrast, verbose model queries hope search results contain *something* helpful—it does not mind reading through a dozen search results. Another reason might be that QA models are worse than humans: for QB questions randomly given to players, 56.58% of the questions are correctly answered by players, while only 44.21% are correctly answered by IRRR.[3]

[3]For questions randomly sampled from HotpotQA, human accuracy (71.43%) is slightly lower than IRRR's 79.02%. We consider this to be due to the synthetic construction of the HotpotQA dataset: it lends itself to straightforward searches. QB better discriminates (by design) between agents' ability.

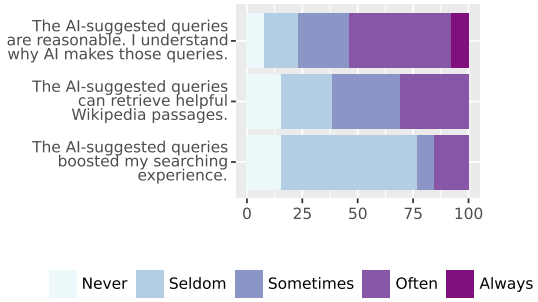

Figure 5: From our post-survey, players' feedback for queries suggested by QA models. Although most players understand why models make those queries, players doubt the utility in improving the search experience.

5.1 Improve Existing Models with Human Actions

Though QA models fail to help humans advance their searches, could the accuracy of the QA models increase if we replace computer queries with humans'? We compare how well IRRR performs under two settings: querying and answering from scratch (**scratch**) v.s. initializing the model's reasoning path from the human reasoning path (**init from human**).

To convert human queries into IRRR's format, given the full action path $A = (q_0, \mathcal{E}_1, q_1, \mathcal{E}_2, \cdots, q_{k-1}, \mathcal{E}_k, a)$ of question Q, for each $0 \leq j \leq k-1$, we trim the action path that ends with query q_j to form a partial human action path $A_j = (q_0, \mathcal{E}_1, q_1, \mathcal{E}_2, \cdots, q_j)$. We initialize the reasoning path R with $R = (Q)$. Then, we extend the reasoning path with documents from the user. Because IRRR can only look at one document at a time, we need to decide which document to append to the reasoning path R. For each $\mathcal{E}_i$ $(1 \leq i \leq j)$ in action path A_j, if $\mathcal{E}_i \neq \emptyset$, we append the most crucial document $d_i \in \mathcal{E}_i$ in this order: 1) source of player answer 2) source of some query 3) manually recorded by the player as evidence, since they are more likely to lead the model to generate human-like queries and answers. We consider the converted human reasoning path $R_l = (Q, d_1, d_2, \cdots, d_l)$ to be the reasoning path of reasoning step l, where $l \leq j$ since there might be empty $\mathcal{E}_i$. Note that as the special case, we consider $R_0 = (Q)$ and $A_0 = (q_0)$.

We compare the two settings on the question set $\mathcal{Q}_l$, which is the set of questions where partial human actions A_j could be converted to a human reasoning path at reasoning step l $(0 \leq l \leq 2)$.

Questions	Scratch	Init from human
$\mathcal{Q}_0$	44.21%	50.45%
$\mathcal{Q}_1$	38.10%	42.42%
$\mathcal{Q}_2$	27.69%	37.95%

Table 2: Compared to querying from scratch, IRRR answer accuracy greatly increases after initializing from human actions, suggesting models benefit from humans' insights.

Obviously $\mathcal{Q}_2 \subseteq \mathcal{Q}_1 \subseteq \mathcal{Q}_0$. We have converted $|\mathcal{Q}_0| = 1122, |\mathcal{Q}_1| = 462, |\mathcal{Q}_2| = 195$ questions in total. The difficulty of questions in $\mathcal{Q}_2$ is, in general, greater than questions $\mathcal{Q}_0$ since humans use at least three queries for answering the questions in $\mathcal{Q}_2$, while using at least one query for $\mathcal{Q}_0$.

Initializing from human actions significantly improves the accuracy of the final answer (Table 2), outperforming querying from scratch by 10.26% for questions in $\mathcal{Q}_2$. The human queries can unlock reasoning paths that make previously unanswerable questions answerable within three steps. While humans cannot get much from computer queries, the reverse is certainly true. We further qualitatively analyze why human actions are helpful to models.

Better selection of keywords. For questions where IRRR answers correctly with human initialization but fails alone, 91.48% of the first queries are substrings or derived from the question. Models select more keywords (Section 4.2); however, this strategy might fail when the retrieval results are too diffuse. In the last example from Table 1, the first IRRR query retrieves weakly related documents, and IRRR appends a paragraph from "Cultural impact of the Beatles" to the reasoning path. Since IRRR can only use a single search chain, the second and the third query follow previous evidence and retrieves more irrelevant documents. In comparison, the player query "high hopes song" allows IRRR to find "High Hopes (Frank Sinatra song)" and use it as evidence. That paragraph contains key information—the film *A Hole in the Head*—which unlocks the film's director, Frank Capra.

World Knowledge. A small proportion of human queries "improves" the model accuracy because it directly includes the answer or shortcuts to the answer. As an example, the first human query for the question

> The first one of these to be directly observed was obtained by the solution of TBF in an antimony-based superacid.

is "George Olah", a Hungarian-American chemist associated with "superacids". IRRR uses this shortcut to find the answer "carbocations" on the Wikipedia page "George Andrew Olah".

5.2 Design Suggestions for Future Models

Based on the strategic differences between human and QA models, we propose improvements for future query-driven open-domain QA models.

Retriever-Aware Queries. The model should be able to interact with the retrieval system, dynamically refine imperfect queries based on retrieval results, and abandon search chains that cannot lead to the answer. Queries can be refined by deleting and adding words, using search operators (Adolphs et al., 2022a), or adding masks to tokens for dense queries (Zhang et al., 2021). Query refinement can be trained through reinforcement learning (Adolphs et al., 2022a) or supervised learning from a synthetic query reformulation dataset (Adolphs et al., 2022b). If retrieval results are irrelevant to the question, the model should discard the results: $\mathcal{E}_i = \emptyset$, avoiding the introduction of noise for future query generation. Models should be able to dynamically select search engines and specify search sources suitable for each query.

Incorporate Common Sense and World Knowledge. Instead of using substrings from questions and previous evidence as queries, the model should add words and terms to queries just like humans do, either via templates, or using a generative language model (Li et al., 2022). Other methods for incorporating common sense and world knowledge include accessing an external knowledge base (Woods et al., 1972; Harabagiu and Maiorano, 1999) and reasoning over knowledge graphs (Lin et al., 2019; Zhang et al., 2022).

Check Your Work. Models should explicitly check the correctness of candidate answers. A simple yet effective strategy humans use is to directly query the candidate answer and see whether it can retrieve documents related to the question. Previous research also explores answer validation through abduction (Harabagiu and Maiorano, 1999) and via Web information (Magnini et al., 2002).

A model that satisfies the above design principles could be implemented using reinforcement learning with a well-defined environment and reward

function. Such a system would provide flexibility—enabling dynamic query refinement (Huebscher et al., 2022) and abandonment—which are not supported by traditional QA systems. This direction would also solve more complex QA tasks that require planning and balancing, e.g., answering incremental questions (Rodriguez et al., 2021) with fewest clues and fewest searches. The contributions in this paper make this possible; for example, a reinforcement learning agent could be trained from our data by behavior cloning.

6 Related Work

Human Use of Search Engines. Our work is similar to previous research that analyzes the behavior of humans using search engines. (O'Day and Jeffries, 1993) discover that it is crucial to reuse the results from the previous searches to address the information need. (Lau and Horvitz, 1999) evaluate the logs of the Excite search engine and find that each information goal requires 3.27 queries on average. (Jansen et al., 2009; Huang and Efthimiadis, 2009) find that contextual query refinement is a widely used strategy. Queries are refined by incorporating background information and evidence from past search results, which usually include examining result titles and snippets. Our work provides many of the same features as these previous papers but adds neural models to retrieve passages, suggest queries, and extract answers. Our analysis focuses on comparing human and computer search strategies and how they may benefit each other. In addition, our task gamifies the search task using the unique structure of QB questions, which is intended to make the task more challenging and fun.

Question Answering Agents. Previous work has explored agents that issue interpretable text-based queries to a search engine to answer questions. GOLDEN retriever (Qi et al., 2019) generates a query by selecting a span from the reasoning path, and IRRR (Qi et al., 2021) further advances the GOLDEN retriever by allowing queries to be any subsequence of the reasoning path. Adolphs et al. (2022a) train an agent using reinforcement learning to interact with a retriever using a set of search operators. WebGPT (Nakano et al., 2021) is a large language model based on GPT-3 (Brown et al., 2020) that searches queries in natural language and aggregates information from multiple web pages to answer *open-ended* questions.

Alternative Models. While our work only compares human search strategies with computer systems that answer questions by searching text-based queries, modern retrievers directly compose vector queries (Karpukhin et al., 2020; Xiong et al., 2021; Zhao et al., 2021), hop through different documents by following structured links (Asai et al., 2020; Zhao et al., 2020), or resolve coreference (Chen et al., 2021). However, we consider that vector-based queries are confusing black boxes for human players. Thus, computer systems using vector-based queries could hardly collaborate with humans. Players say they use Wikipedia links to directly jump to other Wikipedia pages. Thus, following these structured links or resolving coreference could be modeled by query-generation systems: if a user clicks on a Wikipedia link, that could be a part of the next query.

7 Conclusion

Open-domain and multi-hop QA is an important problem for both humans and computers. To compare how humans and computers search and answer complex questions, our interface collects human question answering data as agents search with traditional and neural search engines alongside question answering models that suggest queries and answers. Humans often use shorter queries, apply dynamic search chains, and use world knowledge. Future QA models should have the ability to generate novel queries, "discard" irrelevant results, and explicitly check answers. Moreover, computer agents for QA should also be able to use diverse retrievers to find evidence to answer questions, learning from the insights found in human data. With an agent trained on our data, we could have the "best of both worlds" to combine the ingenuity and tacit knowledge of humans with an indefatigable agent with access to all the world's information.

Acknowledgements

We thank Michelle Yuan, Shi Feng, Chenglei Si and the anonymous reviewers for their insightful feedback. We thank Tsinghua University (Wanrong He) and the National Institute of Standards and Technology (Andrew Mao) for fellowships that supported the authors' research experience. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the researchers and do not necessarily reflect the views of the funders. We thank the Cheater's Bowl participants for

supporting this work by providing their search session data. Boyd-Graber is supported by NSF Grant IIS-1822494 and by ODNI, IARPA, via the BETTER Program contract #2019-19051600005. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies, either expressed or implied, of ODNI, IARPA, or the US Government. The US Government is authorized to reproduce and distribute reprints for governmental purposes notwithstanding any copyright annotation therein.

Limitations

The first limitation of this work is that we only provide Wikipedia as the single source for information retrieval because Wikipedia is the common retrieval source used in open-domain QA models; hence we failed to directly illustrate the human behavior of searching over multiple sources. The second limitation is that for human-AI collaboration, we mainly use IRRR and GOLDEN retriever as the representative of AI models since they are state-of-the-art multi-hop QA models that generate text-based queries. QA models that use different strategies could be further explored and compared with human strategies.

Ethical Concerns

We took steps to ensure our data collection process adhered to ethical guidelines. Our study was IRB-approved. We paid players who actively participated in the gamified data collection process ($130 for top players and $25 for the raffle). We got feedback from the online trivia community before and after launching our game (Appendix A). We release our data to the public domain.

References

Leonard Adolphs, Benjamin Börschinger, Christian Buck, Michelle Chen Huebscher, Massimiliano Ciaramita, Lasse Espeholt, Thomas Hofmann, Yannic Kilcher, Sascha Rothe, Pier Giuseppe Sessa, and Lierni Sestorain. 2022a. Boosting search engines with interactive agents. *Transactions on Machine Learning Research*.

Leonard Adolphs, Michelle Huebscher, Christian Buck, Sertan Girgin, Olivier Bachem, Massimiliano Ciaramita, and Thomas Hofmann. 2022b. Decoding a neural retriever's latent space for query suggestion. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, Online and Abu Dhabi. Association for Computational Linguistics.

Bryce Allen. 1991. Topic knowledge and online catalog search formulation. *The Library Quarterly*, 61(2).

Akari Asai, Kazuma Hashimoto, Hannaneh Hajishirzi, Richard Socher, and Caiming Xiong. 2020. Learning to Retrieve Reasoning Paths over Wikipedia Graph for Question Answering. In *International Conference on Learning Representations*.

Anne Aula, Natalie Jhaveri, and Mika Käki. 2005. Information search and re-access strategies of experienced web users. In *Proceedings of the World Wide Web Conference*.

Steven Bird, Ewan Klein, and Edward Loper. 2009. *Natural Language Processing with Python*. O'Reilly Media, Inc.

Jordan Boyd-Graber, Brianna Satinoff, He He, and Hal Daumé III. 2012. Besting the quiz master: Crowdsourcing incremental classification games. In *Proceedings of Empirical Methods in Natural Language Processing*.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D 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 Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language Models are Few-Shot Learners. In *Advances in Neural Information Processing Systems*.

Jifan Chen, Shih-ting Lin, and Greg Durrett. 2021. Multi-hop Question Answering via Reasoning Chains. *ArXiv*, 1910.02610.

Clinton Gormley and Zachary Tong. 2015. *Elasticsearch: the definitive guide: a distributed real-time search and analytics engine*. O'Reilly Media, Inc.

Sanda Harabagiu and Steven Maiorano. 1999. Finding Answers in Large Collections of Texts: Paragraph Indexing W Abductive Inference. *Proc. AAAI Fall Symposium on Question Answering Systems*.

Jeff Huang and Efthimis N. Efthimiadis. 2009. Analyzing and evaluating query reformulation strategies in web search logs. In *Proceedings of the ACM International Conference on Information and Knowledge Management*.

Michelle Chen Huebscher, Christian Buck, Massimiliano Ciaramita, and Sascha Rothe. 2022. Zero-shot retrieval with search agents and hybrid environments. *ArXiv*, 2209.15469.

Bernard J. Jansen, Danielle L. Booth, and Amanda Spink. 2009. Patterns of query reformulation during web searching. *Journal of the American Society of Information Science Technology*, 60(7).

Bernard J. Jansen, Amanda Spink, and Tefko Saracevic. 2000. Real life, real users, and real needs: a study and analysis of user queries on the web. *Information Processing & Management*, 36(2).

Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. 2020. Dense passage retrieval for open-domain question answering. In *Proceedings of Empirical Methods in Natural Language Processing*.

Tessa Lau and Eric Horvitz. 1999. Patterns of search: Analyzing and modeling web query refinement. In *International Conference on User Modeling*.

Shuyang Li, Mukund Sridhar, Chandana Satya Prakash, Jin Cao, Wael Hamza, and Julian McAuley. 2022. Instilling type knowledge in language models via multi-task QA. In *Conference of the North American Chapter of the Association for Computational Linguistics*.

Bill Yuchen Lin, Xinyue Chen, Jamin Chen, and Xiang Ren. 2019. KagNet: Knowledge-aware graph networks for commonsense reasoning. In *Proceedings of Empirical Methods in Natural Language Processing*.

Bernardo Magnini, Matteo Negri, Roberto Prevete, and Hristo Tanev. 2002. Is it the right answer? Exploiting web redundancy for answer validation. In *Proceedings of the Association for Computational Linguistics*.

Reiichiro Nakano, Jacob Hilton, Suchir Balaji, Jeff Wu, Long Ouyang, Christina Kim, Christopher Hesse, Shantanu Jain, Vineet Kosaraju, William Saunders, Xu Jiang, Karl Cobbe, Tyna Eloundou, Gretchen Krueger, Kevin Button, Matthew Knight, Benjamin Chess, and John Schulman. 2021. WebGPT: Browser-assisted question-answering with human feedback. *ArXiv*, 2112.09332.

Vicki L. O'Day and Robin Jeffries. 1993. Orienteering in an information landscape: How information seekers get from here to there. In *International Conference on Human Factors in Computing Systems*.

Peng Qi, Haejun Lee, Tg Sido, and Christopher Manning. 2021. Answering Open-Domain Questions of Varying Reasoning Steps from Text. In *Proceedings of Empirical Methods in Natural Language Processing*.

Peng Qi, Xiaowen Lin, Leo Mehr, Zijian Wang, and Christopher D. Manning. 2019. Answering complex open-domain questions through iterative query generation. In *Proceedings of Empirical Methods in Natural Language Processing*, Hong Kong, China. Association for Computational Linguistics.

Pedro Rodriguez, Shi Feng, Mohit Iyyer, He He, and Jordan Boyd-Graber. 2021. Quizbowl: The Case for Incremental Question Answering.

Daniel M. Russell. 2019a. *The Joy of Search: A Google Insider's Guide to Going Beyond the Basics*. MIT Press.

Daniel M. Russell. 2019b. *The Mystery of the Parrotfish, or Where Does That White Sand Really Come From? How to Triangulate Multiple Sources to Find a Definitive Answer*. The MIT Press.

Richard S. Tedlow. 1976. Intellect on Television: The Quiz Show Scandals of the 1950s. *American Quarterly*, 28(4).

Keenan Trotter. 2013. Harvard and the Question of Quiz Bowl Cheating. *The Atlantic*.

William A. Woods, Ronald M. Kaplan, and Bonnie Nash-Webber. 1972. The Lunar Sciences Natural Language Information System: Final report. Technical Report 2378, Bolt, Beranek, and Newman, Inc., Cambridge, MA.

Wenhan Xiong, Xiang Li, Srini Iyer, Jingfei Du, Patrick Lewis, William Yang Wang, Yashar Mehdad, Scott Yih, Sebastian Riedel, Douwe Kiela, and Barlas Oguz. 2021. Answering Complex Open-Domain Questions with Multi-Hop Dense Retrieval. In *Proceedings of the International Conference on Learning Representations*.

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William Cohen, Ruslan Salakhutdinov, and Christopher D. Manning. 2018. HotpotQA: A dataset for diverse, explainable multi-hop question answering. In *Proceedings of Empirical Methods in Natural Language Processing*.

Chen Zhang, Yuxuan Lai, Yansong Feng, and Dongyan Zhao. 2021. Extract, integrate, compete: Towards verification style reading comprehension. In *Findings of the Association for Computational Linguistics: EMNLP*.

Xikun Zhang, Antoine Bosselut, Michihiro Yasunaga, Hongyu Ren, Percy Liang, Christopher D Manning, and Jure Leskovec. 2022. GreaseLM: Graph REASoning enhanced language models. In *Proceedings of the International Conference on Learning Representations*.

Chen Zhao, Chenyan Xiong, Jordan Boyd-Graber, and Hal Daumé III. 2021. Multi-step reasoning over unstructured text with beam dense retrieval. In *Conference of the North American Chapter of the Association for Computational Linguistics*.

Chen Zhao, Chenyan Xiong, Corby Rosset, Xia Song, Paul Bennett, and Saurabh Tiwary. 2020. Transformer-XH: Multi-evidence Reasoning with Extra Hop Attention. In *Proceedings of the International Conference on Learning Representations*.

A Player Feedback Survey

We gathered valuable feedback from our players about the data collection experiment, both to understand our human strategies, and improve our system to be more enjoyable. We sent them a questionnaire with the following questions:

- Which search engine do you prefer?
- How do you like these search engines?
- How often do you search for things from these sources? (1 to 5):
  - Original question
  - Wikipedia page (resulted from previous search)
  - AI-suggested queries
  - My own knowledge about the question
- Please rate how much you agree with each of the statements (1 to 5):
  - The AI-suggested queries boosted my searching experience.
  - The AI-suggested queries can retrieve helpful Wikipedia passages.
  - The AI-suggested queries are reasonable. I understand why AI makes those queries.
- Select the search strategies you have applied. (List of strategies)
  - Search (multiple) keywords/specialized terms
  - Utilize the links in Wikipedia pages, directly jump to another page
  - Use world knowledge about the question/domain
  - Learn domain-specific knowledge from the results, and use them in future search
  - Add proper words to restrict the range of results (for example, the subject category like "philosophy", "chemistry", name of the topic, ...)
  - Try name variants, e.g., Matthew C Perry → M. C. Perry
  - Refine the previous query if it doesn't yield any helpful results
  - At the beginning/when unclear, make simple & broad query (e.g. a single noun or phrase)
  - Search candidate answer to verify its correctness
  - Chain of searches: next query is based on previous search results
  - Parallel searching chains: use multiple separate search chains.
  - Search in multiple search engines.
  - Search in multiple languages
- Could you tell us more about your search strategy, and why you use it?
- What feature would you like to see included in this app? Is there a feature that will make finding answers easier, but we don't have it yet?
- Any other feedback for Cheater's Quizbowl?

Overall we received 13 responses.

The large majority (13) of respondents preferred ElasticSearch over DPR (2), with most saying ElasticSearch better met their expectations: the Wikipedia page in their queries always ranked top. The two players who also like DPR consider DPR can retrieve what they are looking for when using natural language queries.

As is shown in Figure 6, players mostly queries from the original question, and also from the previous retrieval results. Players seldomly use queries suggested by the QA models.

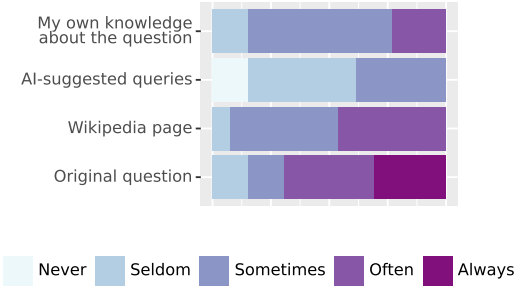

Figure 6: Source of player queries. Respondents reported that they seldomly use queries suggested by the QA models.

Most respondents didn't find the AI suggested queries useful, but most thought they were sensible, and sometimes retrieved relevant passages (Figure 5).

The majority of respondents used the following strategies: clicking on Wikipedia links, refining the previous query, searching the candidate answer

to validate it, creating a search chain where the next query is based on the previous passages, using multiple search chains, and using world knowledge. All strategies listed above received at least two respondents claiming that they have used it.

People also reports diverse strategies they have applied. Interesting responses includes

> I think the inclination toward keyword search has to do with the desire for "the" answer rather than "an" answer. I definitely use natural language queries in normal searches, but usually when I am looking for a subjective answer, or a variety of options. I might google something like "how often should I wash my car" or "what's the best teapot" - questions that have possible answers, but not a single objectively correct answer. In those cases I'm happy to sort through many responses to synthesize an answer. But in Quizbowl (and especially in this case given the time/search constraints) I don't want to spend time typing a long query, or paraphrasing what's in the question, and I definitely don't want to risk getting answers that are contradictory or ambiguous. The goal is to search something specific and uniquely identifying that leads clearly to a single correct answer and keywords just seem so much safer for that goal.

> Check the AI suggestions, and use one of them if they seem sensible, or type my own. Then develop it from there, based on the top results and seeing if there are any leads.

> I used different strategies for different questions. I figured out quickly that the AI-generated queries were mostly not helpful for me unless they were one person's name. In those cases I found myself scanning biographical entries from the beginning and eventually getting a clue that would help me find an answer. Adding a subject category like philosophy or chemistry in the initial search was often useful. Questions about the content of literary texts and visual art were really difficult to search; I could get closer to the answer but not all the way there.

B Implementation Detail

Here we provide the implementation details for the Cheater's Bowl interface.

B.1 ElasticSearch

We set up ElasticSearch with minor modifications from (Qi et al., 2021). We use the ElasticSearch version of 6.8.2. The index is built based on the English Wikipedia dumped on Aug 1st, 2020. We first split each Wikipedia page into paragraphs, and then index individual paragraphs (including both the text and links).

B.2 Pretrained Models

The pretrained IRRR model we used in our experiments can be downloaded from https://nlp.stanford.edu/projects/beerqa/irrr_models.tar.gz, and the pretrained GOLDEN model can be downloaded through the shell script https://github.com/qipeng/golden-retriever/blob/master/scripts/download_golden_retriever_models.sh. The pretrained DPR model we used for the search engine can be downloaded from https://dl.fbaipublicfiles.com/dpr/checkpoint/retriever/single/nq/hf_bert_base.cp.