Ready Player One! Eliciting Diverse Knowledge Using A Configurable Game

Agathe Balayn*, Gaole He*, Andrea Hu*, Jie Yang, and Ujwal Gadiraju
Web Information Systems, Delft University of Technology
Delft, The Netherlands
{a.m.a.balayn,g.he,j.yang-3,u.k.gadiraju}@tudelft.nl,hjc3299@gmail.com

ABSTRACT

Access to commonsense knowledge is receiving renewed interest for developing neuro-symbolic AI systems, or debugging deep learning models. Little is currently understood about the types of knowledge that can be gathered using existing knowledge elicitation methods. Moreover, these methods fall short of meeting the evolving requirements of several downstream AI tasks. To this end, collecting broad and tacit knowledge, in addition to negative or discriminative knowledge can be highly useful. Addressing this research gap, we developed a novel game with a purpose, '**FindItOut**', to elicit different types of knowledge from human players through easily configurable game mechanics. We recruited 125 players from a crowdsourcing platform, who played 2430 rounds, resulting in the creation of more than 150k tuples of knowledge. Through an extensive evaluation of these tuples, we show that **FindItOut** can successfully result in the creation of plural knowledge with a good player experience. We evaluate the efficiency of the game (over 10× higher than a reference baseline) and the usefulness of the resulting knowledge, through the lens of two downstream tasks — *commonsense question answering* and the identification of *discriminative attributes*. Finally, we present a rigorous qualitative analysis of the tuples' characteristics, that informs the future use of **FindItOut** across various researcher and practitioner communities.

CCS CONCEPTS

• **Human-centered computing** → **Collaborative and social computing**; **Interface design prototyping**.

KEYWORDS

GWAP, knowledge elicitation, discriminative knowledge, neuro-symbolic AI, commonsense, human computation

ACM Reference Format:
Agathe Balayn*, Gaole He*, Andrea Hu*, Jie Yang, and Ujwal Gadiraju . 2022. Ready Player One! Eliciting Diverse Knowledge Using A Configurable Game. In *Proceedings of the ACM Web Conference 2022 (WWW '22), April 25–29, 2022, Virtual Event, Lyon, France.* ACM, New York, NY, USA, 11 pages. https://doi.org/10.1145/3485447.3512241

* Equal Contribution.

This work is licensed under a Creative Commons Attribution-NonCommercial-ShareAlike International 4.0 License.

WWW '22, April 25–29, 2022, Virtual Event, Lyon, France
© 2022 Copyright held by the owner/author(s).
ACM ISBN 978-1-4503-9096-5/22/04.
https://doi.org/10.1145/3485447.3512241

1 INTRODUCTION

With the proliferation of AI and machine learning across domains, access to knowledge is an ubiquitous necessity [40, 50]. For instance, years ago knowledge was found to be useful for building automated agents that reason over commonsense facts [39]. This necessity is now resurfacing with the development of machine learning techniques for diverse use-cases [10]. Knowledge can be used to assess the validity of the "knowledge patterns" acquired by machine learning models and highlighted by recent explainability works [33, 34] for various inference tasks [20, 23]. In recent neuro-symbolic AI works, knowledge is integrated into the models [13, 21] to facilitate the learning of inference mechanisms that are more accurate since they do not rely solely on potentially biased statistical data patterns.

Knowledge engineering is the area of research that focuses on developing methods to gather knowledge [37]. Knowledge is gathered by interrogating humans through simple interfaces or complex interactions such as games with a purpose, by mining existing textual resources, or by logically reasoning about known facts to infer new ones [16, 50]. In light of the renewed need for knowledge, we have identified three important gaps pertaining to these knowledge elicitation methods, that we aim to address in this work.

• Our understanding of the *type of knowledge* that can be gathered through these methods remains shallow. Knowledge can be categorized using different typologies of qualities depending on the domain and its envisioned use. It varies from explicit to tacit, from general to specific, from conceptual to situational, from shallow to deep, from commonsense to expertise, etc. Yet, previous works have not provided an in-depth characterization of the knowledge they collected. This might be a barrier to leveraging such knowledge in the context of AI tasks. For example, consider the question "*What does one gain from getting a divorce?*", and the choices –bankruptcy, sadness, depression, tears and freedom. While the first four seem highly relevant to "divorce", the mention of "gain" indicates positivity, hence "freedom" is the right answer. Here, it is important to associate "gain" with something positive, which humans are capable of doing tacitly. Tacit and commonsense knowledge –"knowledge about the everyday world that is possessed by all people"[25], that has the qualities of being shared by multiple persons, and of being fundamental, implicit, large-scale, open-domain [50]– has been heralded as a pivotal ingredient for future AI systems [26].

• Gathered knowledge remains limited and *incomplete* [22], leading to errors in certain tasks. Elicitation methods largely facilitate the creation of *generative* knowledge, but neither *discriminative*, nor *negative* knowledge –despite the fact that novel AI tasks require such knowledge, *e.g.,* for discarding erroneous AI models [2, 3, 22]. Discriminative knowledge allows to distinguish between two concepts (*e.g., octopus*, contrary to *fish, do not have fins*) — as opposed

to generative knowledge that qualifies a single concept. Negative knowledge informs on the invalidity of a tuple to characterize a concept or two compared concepts (*e.g., man* is **not** a *profession*).

• Leveraging human intelligence and commonsense knowledge can allow to collect targeted knowledge beyond what is found in existing resources. However, owing to a lack of understanding of types of knowledge that can be elicited from humans (or online crowd workers), and the concomitant breadth of knowledge, typical knowledge acquisition methods are not readily configurable to meet varying requirements (*e.g.,* knowledge tacitness, specificity).

We position our work in the context of knowledge elicitation techniques involving the crowd [16, 45, 50]. Herein, we draw inspiration from prior work in the realms of games with a purpose (GWAPs), which have shown promise in collecting diverse knowledge in an efficient manner. Popular GWAPs, such as the ESP game [46], Peekaboom [49], and Phetch [47] have provided evidence to show the efficiency of this approach, and its flexibility (*e.g.,* use of gamification and mechanics such as taboo words to tune the type of collected data). Combined with the development of crowd computing frameworks [11], GWAPs can allow for large-scale acquisition of knowledge while engaging humans using different incentives.

To the best of our knowledge, however, no GWAP has been developed or proposed to gather discriminative or negative knowledge. Hence, we first design and implement a novel GWAP called '**FindItOut**', to elicit plural knowledge from players. We then characterize the diversity of knowledge that can be collected using **FindItOut**, and the utility of such knowledge in relevant AI tasks. We highlight the suitability of **FindItOut** in encouraging players to combine explicit knowledge and externalize relevant tacit knowledge. Finally, we demonstrate the efficiency of the game subject to different parameters. We make the following contributions:

- A novel configurable GWAP[1] that facilitates the collection of positive and negative, generative and discriminative knowledge, while facilitating an enjoyable player experience.
- A structured set of dimensions through which one can characterize knowledge collected through user interactions.
- A characterization of the types and quality of knowledge that can result from using **FindItOut** and paid online crowdsourcing.
- An extensive evaluation of the throughput and utility of the game for two distinct AI tasks.

Our results demonstrate that **FindItOut** is highly efficient in obtaining tacit, discriminative and negative knowledge — absent from existing knowledge bases. We also show that the configurability of the game allows to elicit knowledge that can be particularly useful for AI tasks like commonsense question answering and identification of discriminative attributes.

2 BACKGROUND & RELATED LITERATURE

2.1 Knowledge As a Topic of Enquiry

In the Social Sciences. Different typologies of knowledge have emerged [31]. One of the most common ones considers explicitness. Explicit knowledge "can be articulated into formal language [.. and] can also be readily transmitted to others."[8]. Conversely, tacit knowledge is hard to articulate. It "consists of informal, hard-to-pin-down skills, [..] mental models, beliefs, and perspectives so ingrained that we take them for granted and cannot easily articulate them" [29].

There is a higher chance that explicit knowledge already resides in available knowledge bases, as opposed to tacit knowledge [18]. The game we propose involves human players and pushes them to formulate statements about concepts they might not immediately think of. We therefore hypothesise (and evaluate) that our game allows to collect tacit knowledge in addition to the explicit kind.

The distinction between tacit and explicit knowledge has primarily been used to formalise the process of knowledge creation in organizations [29]. Particularly, *combination* [28] is the process of synthesizing explicit knowledge from the combination of previous explicit knowledge. Our game realizes this by synthesizing explicit knowledge about diverse concepts into a single knowledge repository. *Externalization* [28] is the process of creating explicit knowledge from tacit knowledge, often using interviews and questionnaire with experts, or expert's self-analysis [27]. In our work, we evaluate the extent to which our GWAP, **FindItOut**, can support and operationalize externalization through the game mechanics.

In Computer Science. Recent AI inference tasks describe *discriminative knowledge* in contrast to *generative knowledge*. While generative knowledge broadly corresponds to information about different entities, discriminative knowledge allows to identify differences between these entities, which "allow to grasp subtle aspects of meaning [.. and] contribute to the progress in computational modeling of meaning" [22]. Recent works [2, 3] on knowledge inference under the open-world assumption also discuss the importance of *negative knowledge*. It may enhance knowledge bases for knowledge exploration and question answering. Biswas et. al [5] also propose to leverage negative statements as clues to help players find answers to specific questions. Concomitant with the growing interest in these types of knowledge, **FindItOut** is the first GWAP that directly collects discriminative and negative knowledge, which can always be turned into generative one via simple post-processing.

2.2 GWAPs for Knowledge Elicitation

Games with a purpose (GWAP) are used to collect large quantities of knowledge efficiently from the crowd [45]. They have been shown to perform well to collect certain types of knowledge.

Multiplayer GWAPs. Verbosity [48] was the first GWAP proposed for collecting commonsense knowledge. It is a two-player, Taboo-inspired, collaborative game, where a narrator player gives hints to a guesser player who should guess the word the narrator is hinting at. It uses a scoring system to incentivize players to provide the most relevant inputs. A single-player version also exists in order to validate the collected knowledge. The hints have a template format with a relation to fill in with additional words. Common Consensus [24] is a competitive game inspired from FamilyFeud, that collects goal-specific knowledge. It generates questions based on a list of goals and a list of template-questions, and players enter as many possible answers (single words) as possible. Scores are computed based on the number of players with the same answers.

Single-player GWAPs. RobotTrainer [32] is a game, that collects knowledge rules, ranks their appropriateness, and evaluates their validity. For this, it is organized in three levels, where players get to write template-based rules that should serve to answer a question

[1] https://github.com/delftcrowd/FindItOut

about a given short story, or evaluate these rules. It is shown to provide similar results to non-game based interactions, but with more engagement of the users. The 20 Questions game [42] requires the player to think about a concept, and the game sequentially generates a list of 20 relation-template based questions to try guessing the concept, questions that the player should answer truthfully. Despite a simple design, players were found to enjoy this game more than a simple template-based input system. The Concept Game [15] similarly generates rules that a player is asked to verify, in order to reduce the cognitive load of players generating assertions.

Other games have been proposed such as Virtual Pet, Rapport, Guess What?!, OntoProto, SpotTheLink [37], that ask players to agree on the relation between concepts, to guess concepts described by other concepts, or to answer questions to extract knowledge.

In comparison to existing GWAPs: (a) **FindItOut** by design, has a higher throughput than previous games. It operationalizes the idea of making both questions and answers relevant to the creation of knowledge. This leads to collect more knowledge in comparison to the aforementioned two-player games, since the two players contribute distinct tuples of knowledge simultaneously, contrary to the other games where players interactions allow for the creation of a single knowledge tuple. (b) **FindItOut** is the only game that directly allows to collect discriminative and negative knowledge. Previous games require either to directly input concepts in relation to a pre-existing characteristic, or to fill in template. They do not leave the space for negative inputs, which also removes the opportunity to indirectly elicit discriminative knowledge. (c) The knowledge that **FindItOut** elicits is, by design, more diverse. While it re-uses the previous ideas of relation templates to fill in, and of scoring systems, it varies from 20 Questions and Common Consensus in that the knowledge it creates is more varied since the rules within the templates are human-generated, and richer than single words (association of relation and up to 5 words).

2.3 Elicitation through Crowd Interactions

Besides GWAPs, other interactive methods [50] exist for knowledge elicitation. A fundamental feature of **FindItOut** is its question answering workflow, which is inspired from the offline game *Guess Who?*, and from crowdsourcing frameworks such as CuriousCat [6], that collects contextual commonsense knowledge, by asking questions to crowd workers that refer to their current environment (*e.g.,* size of a restaurant they are present in). Cosmos QA [17] and Socialiqa [35] are datasets collected by asking crowd workers to formulate questions and answers that require commonsense knowledge, in relation to textual descriptions of everyday situations taken from blogs or prior knowledge bases (*e.g.,* ATOMIC). We draw inspiration from these works and incentivize crowd workers to formulate questions through the game mechanics.

3 DIVERSE KNOWLEDGE EXTRACTION

To elicit and collect discriminative and generative knowledge, that is both positive and negative, we propose **FindItOut** [4] — a competitive 2-player game inspired by the popular game "Guess Who?'. The functional and non-functional requirements that governed the design of the game are elucidated in the companion page [2].

[2]https://sites.google.com/view/finditout-www22/home

3.1 Knowledge Elicitation

In line with existing knowledge bases, we aim to collect knowledge in the form of relations between concepts.

Generative knowledge. A triple of generative knowledge that we collect corresponds to a concept, a relation and a characterizing input, and takes two possible formats. It can be a *positive triple* +<concept, relation, input> where the input is text entered by players in the game. For instance, `+<teapot,UsedFor,making tea>` indicates that the concept *teapot* can be used for *making tea*. We also collect negative knowledge as *negative triples* —<concept, relation, input> that indicate that the relation and input do not apply to the concept. For instance, `---<teapot,UserFor,running>` indicates that the concept *teapot* cannot be used for *running*.

Discriminative knowledge. We also aim to collect discriminative knowledge. This knowledge is represented by positive quadruples +<concept#1, concept#2, relation, input>, where the relation and its associated input apply to concept#1 but not to concept#2, allowing to discriminate between the two. For instance, `<teapot,shoe, UsedFor,making tea>` indicates that the concept *teapot* is different from the concept *shoe* in that only the teapot can be used for making tea. Negative quadruples instead, mean that the relation and input do not allow to discriminate between the two concepts.

3.2 Game Mechanics of `FindItOut`

Initialisation. At the start of the game, both players are presented with a board of multiple cards, that represent different semantic concepts. Each card shows a picture that illustrates the concept, its name, and its potential definitions when one hovers over the card. Game boards can be configured and laid out based on target requirements. These boards are generated with a greedy approach: once a few initial concepts are retrieved for one board, other related ones are appended to the board, either by searching within the WordNet taxonomy, or by adapting to the task at hand — when one wants to understand the difference between two pre-defined concepts, these two concepts can be added simultaneously).

The game randomly assigns a card on the board to each player as their `IT card`. The main goal for each player is to guess the opponent's `IT card` (before their card is identified) by iteratively asking questions and eliminating the possible candidates based on the opponent's responses. The game difficulty can be configured, affecting the number of cards on the board. Game boards with more cards are expected to be more challenging, since they require players to think of questions that ideally discriminate between more concepts simultaneously. We also expect that these boards push players towards articulating more tacit knowledge.

Taking turns in questioning and answering. To balance out the opportunity to win for both players and following the best practices for knowledge elicitation through GWAPs [14], the two players take turns playing the roles of the `Asker` and the `Replier`.

Let Player One be the `Asker` for a given turn. They are given the choice between two actions: `ASKing` or `GUESSing`. Choosing `ASK` prompts Player One to formulate a question to ask Player Two. Player Two is then asked to answer Player One's question, by selecting one among four choices: "yes", "no", "maybe", "unclear". "Maybe" is an appropriate answer in cases where it is ambiguous whether a relation applies to a concept, or if it applies only under

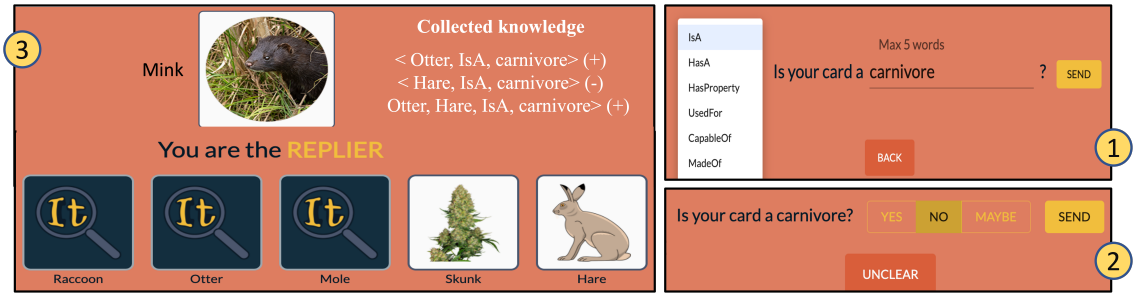


Figure 1: `FindItOut` main interface and workflow. (1) The `Asker` inputs a question. (2) The `Replier` selects an answer. (3) The `Asker` flips relevant cards. Example collected knowledge from this turn is presented in the right top corner of (3) (not in game).

certain conditions. Selecting "unclear" indicates that the question needs to be reformulated by Player One, since Player Two failed to comprehend it. Depending on the answer, Player One flips the cards on the board by clicking on them to eliminate them from contention, and narrow down the possible candidates for Player Two's `IT card`. It is then the end of the turn, and Player Two becomes the `Asker`.

Choosing `GUESS` allows Player One to designate one card on the board as their guess for being Player Two's `IT card`. Then, Player Two is prompted for their own guess, after which the game ends. Player One wins if their guess matches Player Two's `IT card`, otherwise they lose. This action can only be chosen after each player has asked either 2 or 3 questions depending on the easy or difficult game levels respectively. This design choice dissuades players from attempting random guesses that would not contribute to knowledge creation. Figure 1 illustrates this workflow and gameplay.

Question formulation. The questions formulated by the `Asker` follow a template <relation, input>. The *relation* is selected among a pre-defined set of relations, and the input is a natural language proposition to be manually entered by the `Asker` limited to 5 words (for ease of post-processing and to limit the potential for cheating).

We adopt this template-based question answering strategy since previous works have demonstrated their potential efficiency. For instance, the OMCS project [38, 39] identified that structured, relation-based templates are more efficient at collecting rule-type knowledge and the results are more usable than relying entirely on natural language. Thus, by using a combination of template-based and natural language question formulation, **`FindItOut`** provides us with the configurability of tuning the potential target knowledge.

Taboo words. We employ taboo words to ensure that the questions asked by the players are not too simple, and allow to extract useful knowledge. We prevent players from entering natural language inputs that contain words with the same root as the concepts on the board. For example, if a concept on the game board is "bird", a player cannot ask "is my card a bird?". New taboo words can be added over time to prevent collecting redundant knowledge.

3.3 Post-processing the Resulting Knowledge

Extracting knowledge. We process each turn to create knowledge based on heuristics. After receiving a response from the opponent, the asker's flipping card actions provide all information needed to gather new tuples in the form of `(+/--)<card?,relation,input>`, where "`card?`" and sign `(+/--)` are inferred based on whether the card is flipped. Specifically, when the answer to a question is received, the relation and input in the question directly apply to **batch A: reserved cards**, *i.e.,* the batch of cards that were previously unflipped and that remain unflipped, with the sign corresponding to the answer (yes is `+`, and no is `–`). The batch of cards that were previously unflipped and are flipped during the turn (**batch B: flipped cards**) receives the inverse of the sign of the answer. For example, consider the sequence where the question is "does my card have wings", the answer is "no", and then the `Asker` flips the "bird" card, we build the knowledge triple `+<bird, has, wings>`.

Discriminative knowledge is extracted with two concepts in the batch (both A and B) and with a quadruple template. Any concept pair can be gathered to generate discriminative knowledge, which results in $\binom{n}{2}$ (n is the game board size) tuples of knowledge. Considering one concept from each batch allows us to create positive discriminative knowledge, while both concepts from the same batch result in negative discriminative knowledge.

Quality control. It is in the best interest of the `Replier` to lie when replying to a question, such that the `Asker` will be misled (rational game user model [14]). We tackle this issue through our game design. At the end of a game, both players are shown the opponent's `IT card` and their own question history, and can report errors/wrong answers or foul play for any turn. When extracting knowledge from turns, we filter out reported turns automatically and identify outliers for exclusion manually (*e.g.,* players who do not flip cards as required, cheat in the game, ask meaningless questions).

3.4 Technical Implementation

`FindItOut` is implemented as a real-time, responsive web app (see Appendix A.1), for convenience and portability (the game can be served on any platform as long as it supports a web browser). It supports interactions with both voluntary players connecting onto the app, and with players recruited from paid crowdsourcing platforms.

Design choices. The card data are retrieved by querying WordNet for concept definitions, and Google Search for visual representations of the concepts. In the current version of the game, we selected 8 relations, extracted from ConceptNet [25] (IsA, HasA, HasProperty, UsedFor, CapableOf, MadeOf, PartOf, AtLocation) –see Appendix Table 3–, based on their commonality, their applicability to nouns, and adaptedness to the concepts displayed in the game boards. Currently, we propose two game difficulties: easy with 8 cards on the board, and difficult with 16 cards.

4 STUDY DESIGN AND SETUP

FindItOut is designed to be configurable and modular, and thereby to facilitate the elicitation of accurate and diverse knowledge (the concepts we collect knowledge about in this study are chosen to be both abstract and concrete nouns). It is designed to create an enjoyable experience for players, while serving as an efficient means to gather knowledge. These are the objectives we evaluate next.

4.1 Measures and Metrics

We evaluate **FindItOut** through a combination of qualitative and quantitative analyses of the resulting tuples across the two difficulty levels. In identical conditions, no GWAP with crowdsourcing can serve as a directly comparable baseline. Hence, we leverage the standard evaluation lens used for knowledge collection systems [50], in addition to a qualitative analysis of the knowledge and of the enjoyability of the game. These measures are described below:

Efficiency of knowledge collection. We measure the number of tuples (positive and negative triples and quadruples) resulting from the game, as well as the fraction of overlapping knowledge tuples generated by the two players across games and turns. By also considering the average time and number of rounds that a **FindItOut** game lasts as well as its cost, we can measure the throughput and utility of knowledge generation.

Qualities of collected knowledge. We analyze how correct and diverse each resulting tuple is. To this end, we leverage an objective measure — the types of relations that are used during the games, and a subjective measure — we manually rate each resulting tuple on several dimensions (meaningfulness, correctness and multiplicity of interpretations, bias, typicality, specificity, tacitness).

Player experience. We use the player experience inventory questionnaire [1] to evaluate the experience of the players with **FindItOut** and discern the extent to which they enjoy it. Players are asked to complete this questionnaire at the end of all the games that they choose to play in a session. At this stage, we also collect open-ended comments and remarks about the game from players.

4.2 Usefulness of Collected Knowledge

Although the aforementioned measures can help us to understand and quantify the characteristics of the generated knowledge, they do not directly highlight the usefulness of elicited knowledge for concrete AI tasks. To address this, we investigate the usefulness of the generative and discriminative knowledge that we collect, by considering two independent and popular tasks.

Coverage of the 'Discriminative Attribute' task. The discriminative attribute task was introduced as a part of the 2018 SemEval challenge [22], and consists in "predicting" whether one word allows to discriminate between two concepts (*e.g., urine* is a discriminating feature in the word pair of {*kidney, bone*}). This corresponds well with the discriminative knowledge that we collect through **FindItOut**. Hence, we investigate the extent to which populating boards in our game with the concepts of this task and having players interact with these boards allows us to collect such knowledge. We thereby compute the coverage of the elicited knowledge with the discriminative words of the task.

Taking <concept1, concept2, feature> triples from the discriminative attributes (DA) dataset as reference, we first retrieve knowledge tuples extracted from **FindItOut** that share both concepts. Taking these tuples as candidates, we generate reference-candidate pairs to be annotated. We spread the coverage evaluation (whether candidate tuple covers the reference triple) tasks to 5 volunteers, with 10% reference triples in overlap. To make the knowledge tuples readable, we generate statements for both reference and candidates.

Tacit clues for commonsense reasoning. Usefulness of generative knowledge is typically evaluated by measuring the performance gains in subsequent inference tasks, such as question answering which requires rich commonsense knowledge [50]. We generate game boards to extract tuples for a subset of the commonsense question answering (CSQA) benchmark [44], and assess whether the extracted knowledge helps conduct commonsense reasoning.

After generating knowledge tuples, we use SimCSE [12] as a retrieval toolkit to obtain top-k ($k = 5$) relevant candidates for each question-choice pair. To retain candidates which are highly relevant to questions, we filter out those with a similarity less than 0.5. We only retain questions which have at least 10 candidates reserved for all choices, and thereby obtained a subset of 179 questions. Next, we carry out a manual evaluation to label whether candidate knowledge tuples are (1) correct, (2) highly relevant to the question and possibly helpful to infer the answers, or (3) directly confirm the answer or discard a distraction term. Furthermore, we assess whether the collected *useful* knowledge tuples are covered by the primary existing commonsense knowledge base – ConceptNet.

4.3 Participants and Procedure

Players. We recruited participants from the Prolific crowdsourcing platform [30] to play **FindItOut**. All participants were proficient English-speakers above the age of 18 and they had an approval rate of at least 90% on the Prolific platform. We excluded participants from our analysis if they do not flip cards as expected, or represented an outlier in terms of cheating in game (e.g., tell opponent their IT card or give wrong answer quite often) or asking meaningless questions. All participants were rewarded with £2.5, amounting to an hourly wage of £7.5 deemed to be "good" payment by the platform. To encourage participants actively play the game, we rewarded participants with extra bonuses of £0.15 for every win. The players are randomly matched by our system when entering the game, and do not know each other. Players are asked to play 5 mandatory games, three at the easy difficulty level and two at the difficult level. The progressing difficulty allows players to gradually familiarize themselves with the game mechanics. After finishing these five games, the players can play additional games or leave with exit-questionnaire.

Generating Game Boards. For the CSQA task, concepts that appear within a same question are appended to one board (*e.g., {aircraft, school, mexico, battle, human, band, factory, doctor}*, or *{countryside, painting, village, train, ground, mountains, rock, cottage}*. In case of the discriminative attributes (DA) task, concepts from a same triple and from the same semantic field are chosen (*e.g., {mirror, necklace, cigarette, lantern, candle, scarf, lamp, chandelier}*, or *{father, king, daughter, son, prince, uncle, brother, cousin}*).

Concepts from the DA task. To cover as many triples from DA dataset as possible with a limited budget, we only consider triples which contain both frequent concepts (*i.e.,* occur at least 5 times in

positive discriminative triples). Using every concept as a seed, we generated game boards with a greedy search strategy to maximize the triples possibly covered. Considering that game boards of a good diversity can potentially create a better game experience, we filtered out game boards which have overlapping concepts (with a threshold of 2 for easy games and 6 for difficult games). Finally, 41 easy game boards and 22 difficult game boards were generated.

Concepts from the CSQA task. We select the questions from the CSQA dataset [44] that refer to at least 5 meaningful single-word concepts (both question concept and choice concept), resulting in a subset of 864 questions. Similar to the generation of boards for DA dataset, we utilized a greedy search strategy to maximize concepts that occur in the same question to be placed in one game board. With this criteria, multiple questions can be "merged" into one board (see Appendix A.2). Finally, 115 easy game boards and 70 difficult game boards were generated pertaining to the CSQA task.

4.4 Qualitative Assessment of Knowledge

Definition of qualitative dimensions. Owing to the lack of automated and standardized methods to evaluate the quality of knowledge elicited through GWAPs, we carried out a qualitative evaluation of the generated knowledge with respect to the '*correctness*' and '*diversity*' of the knowledge. We manually rated the factual *correctness* of a tuple with either '`correct`', '`incorrect`', or '`not sure`' (when in doubt). We followed an iterative coding process [43] to characterize the *diversity* of the knowledge based on several dimensions informed by related literature in computer science and social science —*correctness, truth, bias, tacitness, typicality, specificity*. Table 1 presents the dimensions used to assess the knowledge tuples. Knowledge is by definition true [31], and it is thus challenging to rate into more than a binary proposition. Hence, we do not use the same Likert-scale dimension as previous works [39], but propose a multi-dimension description of *correctness*.

Annotation procedure. We analyse the qualities of the generative knowledge by selecting and annotating a subset of samples collected from the game boards pertaining to the DA task. We randomly sample 30 difficult games (leading to 1628 generative knowledge tuples), gather the concepts they cover, and then select all knowledge tuples collected through easy games for which the boards include some of the previous concepts (147 games, and 2429 knowledge tuples). The discriminative tuples can be generated from two generative tuples with different signs. Hence, the quality annotation for discriminative tuples is covered by that of generative tuples. 5 authors of this paper annotated 50 generative knowledge tuples selected at random with respect to these dimensions, and refined the codes together until complete agreement was reached. Following this, each of the authors independently annotated **793** tuples, including a common subset of **95** tuples, allowing us to measure the inter-annotator agreement. The Krippendorff's α scores are respectively 0.91 for meaningfulness, 0.37 for correctness (with 0.38 and 0.45 for problematic sign and relation), 0.31 for bias, 0.23 for typicality, 0.39 for specificity (0.51 when using only two values), 0.33 for tacitness (0.43 when using only two values). Disagreement is due to the subjectivity of the task: knowledge and the veracity of a fact vary depending on one's own experience of the world.

5 RESULTS & DISCUSSION

5.1 Game Efficiency

Knowledge quantity. Overall, 255 (164 easy, 91 difficult) and 242 (142 easy, 100 difficult) games were played for the DA and CSQA datasets respectively. This led to collecting 75,491 and 85,923 knowledge tuples. For the DA dataset (and the CSQA dataset respectively), 5.28% (4.39%) of the tuples are generative positive tuples, 6.38% (6.66%) generative negative tuples, 22.8% (20.4%) discriminative positive tuples, and 65.6% (68.5%) discriminative negative tuples.

91.1% of the knowledge tuples pertaining to the DA game boards and 97% w.r.t. CSQA boards consist of unique tuples, while the remaining tuples were generated multiple times across turns or games. On average, easy games lasted 367.2s (*SD=722.3*) in case of DA game boards and 377.8 (*SD=192.3*) for CSQA boards, and corresponded to 3.88 (*SD=1.63*) turns on average for DA game boards, and 4.09 (*SD=1.41*) for CSQA boards. Similarly, difficult games lasted 397.5s (*SD=201.4*) for DA boards —resp. 428.4 (*SD=204.3*) for CSQA boards—, and required 5.69 (*SD=1.98*) —resp. 5.78 (*SD=1.63*)— turns.

Throughput. Overall, for the DA dataset, 13.9 tuples are generated per minute, which is ten times more than Verbosity [48][3].

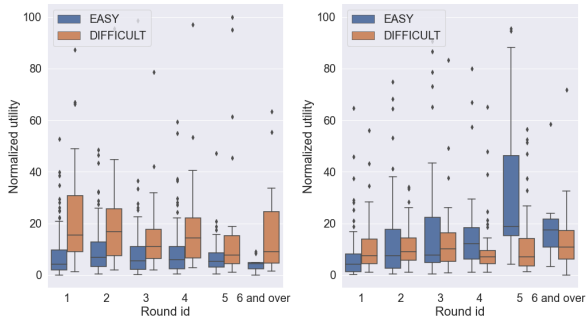
We define the throughput of our game as the number of elicited tuples divided by the time it took (in seconds) to elicit them. In Figure 3 (cf. the Appendix), we report the throughput of our game for both the DA and CSQA tasks, depending on the round of the game, and the type of knowledge tuple elicited. In both cases, the throughput decreases over rounds as there are less uncovered cards in latter rounds, leading to the generation of less tuples when flipping new cards. As expected, the throughput is higher for difficult than easy games, especially for the first rounds of the game. Since there are more cards on the game boards in difficult games, and players are incentivized to ask questions that eliminate as many cards as possible, more knowledge is directly elicited from the early rounds. That is also the reason why the difference between the amounts of discriminative and generative knowledge is higher for these difficult games than the easy ones (a "good" question for the `Asker` leads to an optimum number of flipped/unflipped cards to generate many discriminative tuples). No major difference is observed across datasets as the game mechanics remain the same.

Utility. We compute utility as the fraction of value extracted per unit of time (in seconds) over the cost (in pounds). For the DA dataset, we consider the value extracted to be the number of tuples elicited that are tacit, specific or atypical, as these are tuples that cannot be easily collected from other sources. For the CSQA dataset, we consider the value extracted to be the number of tuples that are correct and relevant for the CSQA task. In Figure 2, we report the normalized utility for the two datasets depending on the round and difficulty of the game. The average utility does not vary significantly over time for the two tasks, albeit with large standard deviations. This is explained by the high variation in the type of knowledge that players elicit through the rounds. Difficult games correspond to a higher utility of **`FindItOut`** for the DA task, while easy games correspond to a higher utility for the CSQA task. In general, larger game boards can aid the generation of more valuable knowledge tuples efficiently due to more cards being included. As CSQA game

[3] According to the approximate numbers reported: *29.47/23.58 = 1.25* tuple per minute.

Table 1: Dimensions on which knowledge tuples are analysed. Labels correspond to the scales used to gather annotations.

	Dimension	Description	Label	Example
Correctness	Validity	A valid tuple is comprehensible [50], and the input is not the result of cheating (*e.g.*, description of visual content on a card).	invalid valid	+(tap, UsedFor, can your card used home), +(mother, HasA, color brown in it) +(camel, AtLocation, in africa)
	Truth	Indicates whether a tuple represents a correct fact.	correct incorrect	+(lamp, HasProperty, makes light) -(mole, IsA, predator), -(squirrel, UsedFor, swimming)
	Meaning(s)	Indicates whether the tuple can have different interpretations (among which at least one is correct), or a single interpretation.	multiple single	+(tower, CapableOf, be used as home) (high-rise building/Eiffel tower) +(avocado, HasProperty, green (most part))
Diversity	Bias	A tuple can be biased due to being true only in certain contexts, since one can biased by their own view of the world.	unbiased biased	+(cucumber, IsA, fruit), -(dishwasher, UsedFor, preserving food) +(crab, HasA, big claws), -(trousers, usedFor, mainly women)
	Typicality	Indicates the perceived typicality of a tuple from one's point of view (so as to acknowledge the subjectivity of certain tuples).	high medium low	+(boat, AtLocation, on water), -(plug, UsedFor, restraining something) +(car, UsedFor, single person), -(finger, AtLocation, on furniture) +(fan, IsA, mostly black in colour), -(aunt, UsedFor, a married person)
	Specificity	Indicates the level of details provided by the input in the tuple. Negative tuples are always specific as there can be an infinite number of negative examples.	high medium low	+(skirt, IsA, typically female clothing), -(tap, UsedFor, restraining sth.) +(zebra, AtLocation, in africa), -(catfish, HasA, shell) +(lamp, HasProperty, makes light)
	Tacitness	Indicates whether one would have a hard time articulating the fact, and the extent to which one tends to readily think of this fact (or its "opposite" fact) when discussing the concept in the tuple	high medium low	+(crab, HasA, red shell when cooked), -(bed, PartOf, kitchen appliance) +(crocodile, AtLocation, jungle), -(avocado, PartOf, group or bunch) +(elephant, IsA, herbivore), -(lion, IsA, herbivore)



(a) Discriminative attribute dataset. (b) CSQA dataset.

Figure 2: Utility of `FindItOut` in relation to each dataset, and computed over different rounds and difficulty levels.

boards are generated based on questions, the smaller the game board the higher is the probability to focus on specific questions. This highlights the benefit of configurability of **`FindItOut`**.

5.2 Analyzing Knowledge Qualities

Below, we report our results for the discriminative attribute dataset.

Correctness. Overall, 95.6% of the generative tuples elicited are meaningful. Among these, 90.6% of the tuples are correct (88.8% and 92.1% respectively for positive and negative tuples). As comparison, Verbosity [48] reports 85% of correct generative tuples elicited. Similarly, 76.2% of the discriminative tuples elicited are correct.

Qualitative study of diversity. As a first indication of the diversity of knowledge types elicited through our game, we investigate the types of relations used by the players. 21.4% of questions employed `IsA`, 20.0% `HasA`, 13.9% `UsedFor`, 13.4% `HasProperty`, 13.1% `CapableOf`, and the other relations in proportions lower than 10%. As each relation corresponds to a different type of information, this shows the diversity of tuples our game collects. A chi-square test of independence to examine the relation between the relations employed by players and the rounds revealed a significant relation, $\chi^2(77, 4235) = 620.59, p < .000$, implying that the relations

Table 2: p-values for Chi-squared tests of independence that were conducted to examine the relation between game rounds and each dimension of the qualitative analysis (†: significant relations).

Level	Correctness	Bias	Typicality	Specificity	Tacitness
All	**3.41e-15**†	**4.55e-08**†	**1.94e-05**†	**1.89e-06**†	**4.89e-04**†
Easy	**5.40e-17**†	**5.22e-04**†	**1.46e-03**†	**1.39e-03**†	2.81e-02
Diff.	**1.22e-05**†	**1.11e-06**†	**2.06e-08**†	**2.24e-03**†	**6.15e-06**†

employed evolved over rounds. In earlier rounds, `IsA` is primarily used as it allows to ask simple, discriminative questions. In later rounds, the frequency of the other relations increases, as more tacit questions need to be asked to distinguish the unflipped cards.

Dimensions. Our qualitative analysis of the elicited knowledge tuples reveal a high diversity in the type of knowledge collected. 86.3% of the tuples are unbiased, 38.3% are highly tacit (21.3% medium), 57.5% highly specific (16.9% medium), 7.98% are atypical. These findings confirm that **`FindItOut`** allow for externalizing tacit knowledge, that is typically not found in existing knowledge bases.

We investigate how the types of knowledge evolved over the rounds, with respect to easy and difficult games, and overall. To this end, we performed Chi-square tests of independence between the annotations of each knowledge dimension and the rounds in the game. To correct for error inflation due to multiple tests, we applied a Bonferroni correction so that the significance threshold of α decreased to $\frac{0.05}{15} = 0.003$. In Table 2, we report the p-values of these tests. Overall, we found that each knowledge dimension evolves across the rounds in which the tuples were elicited. This is consistent across easy and difficult games, except for the tacitness of tuples corresponding to easy games. In Figure 6, we show the percentage of tuples per dimension collected for each round of the game. This indicates the trend of evolution per round. We found that the number of high typicality tuples decreases over rounds, while tuples with high specificity and high tacitness tend to increase after the initial rounds. The reason for such observation is two-fold. After several rounds of a game, reserved concepts are hard to discriminate with general and explicit knowledge. Along the game and its active guessing and thinking mechanisms, players' deeper insights and life experiences are activated/awakened [9].

5.3 Usefulness for AI Tasks

Coverage of discriminative attributes. With 41 easy game boards and 22 difficult game boards generated for the DA dataset, we can cover 3948 triples at most. Due to a limited budget, 55 participants were recruited to play these games, resulting in 3369 triples potentially covered. To filter out noisy reference triples, we manually labelled their validity and found 2987 valid triples (containing 1649 unique concept pairs). These 2987 valid triples are considered as reference. For the annotations of coverage, 5 authors annotated 1102 common samples, and 9808 independent samples. The inter-rater agreement with Krippendorff's α was found to be 0.47, which is reasonable in a subjective task [7]. To evaluate how the generated tuples go beyond the DA dataset, we analyse the correctness of all the candidate tuples (5485) used in coverage annotation. 5 authors annotated 545 common samples, and 4940 independent samples. Inter-rater agreement with Krippendorff's α was found to be 0.43.

For every reference triple, we take all positive discriminative knowledge which have the same concept pairs as candidates. Based on the annotations, we found that 859 (28.8%) of the reference triples are covered. Besides covering a part of the reference triples, we also look into whether the collected candidates can discriminate concept pairs. As manual annotations show, all 1649 concept pairs can be covered with our extracted knowledge, which indicates the extracted knowledge is of high quality and can even go beyond the scope of the DA dataset.

Commonsense question answering. Among 179 questions (every question has five choices), there are 2.82 choices which can find relevant knowledge tuples (correct and possibly useful) per question, and 0.52 choices which can find useful knowledge tuples (correct and can confirm the answer or discard a distraction term). To further verify the usefulness of our extracted knowledge, we find that 20 knowledge tuples (most are tacit knowledge) among 96 unique useful ones (see Section 4.2 last paragraph) are not covered by ConceptNet 5.5. This further verifies the usefulness and necessity of tacit knowledge extracted from **FindItOut**. As all game boards for CSQA subset are only played once (due to a limited budget), we argue that with increased redundancy on the game boards, even more useful knowledge can potentially be elicited.

As shown by previous work [18], existing large-scale commonsense knowledge bases (*e.g.,* ConceptNet [41] and CSKG [19]) are not capable of supporting commonsense reasoning. **FindItOut** fills this gap, by generating both tacit and negative knowledge that is absent from these knowledge bases. Besides reasoning, this negative knowledge can also be leveraged in the future to discard ridiculous inferences and inference mechanisms from machine learning models, which contrast with human commonsense and ethics. This is of great potential to provide trustworthy and robust AI services.

5.4 Player Experience & Enjoyability

Based on our findings from the player experience inventory questionnaire, the main functionalities of **FindItOut** were well understood and appreciated by players. On average players rated the functional consequences (*i.e.,* "the immediate experiences as a direct result of game design choices") with >1 on a scale of -3 to 3. The ease of control and clarity of goals were the best rated dimensions by the players. These highly-rated functional consequences translated into highly rated psychosocial consequences (*i.e.,* "the second-order emotional experiences, such as immersion or mastery") as well, with an average rating per dimension always above 0. This shows that **FindItOut** was enjoyed by players, arose their curiosity by prompting them to think of topics (differences between concepts) that they probably do not typically think of.

5.5 Caveats and Limitations

Considering that game boards play an instrumental role in shaping the nature of the elicited knowledge, it is important that knowledge requirements are translated well into populating the game boards with concepts. To increase the diversity in knowledge, increased redundancy between game boards is required. In this work, we did not explore how **FindItOut** can be extended to the voluntary player contexts where game elements will play an important role. To generate useful and correct knowledge from **FindItOut** automatically, further mechanisms need to be developed to avoid costs entailing human annotations.

6 CONCLUSIONS AND FUTURE WORK

In this paper, we developed a configurable game **FindItOut** to elicit plural knowledge from human players. We evaluate and demonstrate the efficiency of the game, the enjoyable player experience it facilitates, the utility and usefulness of the resulting knowledge, through two downstream AI tasks — commonsense question answering and the identification of discriminative attributes. Results show that our game can generate high-quality discriminative knowledge which goes beyond an existing frame of reference. More importantly, **FindItOut** can generate tacit and negative knowledge which is absent from most mainstream commonsense knowledge bases. **FindItOut** can be easily configured to suit diverse requirements of downstream AI tasks by varying seed concepts, difficulty levels, size of the game boards, the relation sets used for populating question templates, the admissible length of the natural language input from players, using text or image modes, expanding the taboo words that players cannot enter, among other features.

Currently, we only focus on eliciting discriminative and tacit knowledge. Our approach however, can be extended to obtain other types of knowledge and even deeper and contextual insights from human players. In the future, we will also consider enhancing the capability to collect task-specific knowledge, and explore the effectiveness of **FindItOut** when more redundancy is available. While positive knowledge is widely adopted and well studied in existing literature, negative knowledge and discriminative knowledge have not been thoroughly discussed. In the future, we will delve into organizing negative and discriminative knowledge into knowledge bases and explore their usage.

ACKNOWLEDGMENTS

This work was partially supported by the Delft Design@Scale AI Lab, the 4TU.CEE UNCAGE project, and the HyperEdge Sensing project funded by Cognizant. We made use of the Dutch national e-infrastructure with the support of the SURF Cooperative using grant no. EINF-1806. We finally thank all participants from Prolific.

REFERENCES

[1] Vero Vanden Abeele, Katta Spiel, Lennart Nacke, D Johnson, and K Gerling. 2020. Development and validation of the player experience inventory: A scale to measure player experiences at the level of functional and psychosocial consequences. *International Journal of Human-Computer Studies* 135 (2020), 102370.

[2] Hiba Arnaout, Simon Razniewski, Gerhard Weikum, and Jeff Z. Pan. 2021. Negative Knowledge for Open-world Wikidata. In *Companion of The Web Conference 2021, Virtual Event / Ljubljana, Slovenia, April 19-23, 2021*, Jure Leskovec, Marko Grobelnik, Marc Najork, Jie Tang, and Leila Zia (Eds.). ACM / IW3C2, 544–551.

[3] Hiba Arnaout, Simon Razniewski, Gerhard Weikum, and Jeff Z. Pan. 2021. Wikinegata: a Knowledge Base with Interesting Negative Statements. *Proc. VLDB Endow.* 14, 12 (2021), 2807–2810.

[4] Agathe Balayn, Gaole He, Andrea Hu, Jie Yang, and Ujwal Gadiraju. 2021. FindItOut: A Multiplayer GWAP for Collecting Plural Knowledge. (2021).

[5] Aditya Bikram Biswas, Hiba Arnaout, and Simon Razniewski. 2021. Neguess: Wikidata-entity guessing game with negative clues. (2021).

[6] Luka Bradeško, Michael Witbrock, Janez Starc, Zala Herga, Marko Grobelnik, and Dunja Mladenić. 2017. Curious Cat–Mobile, Context-Aware Conversational Crowdsourcing Knowledge Acquisition. *ACM Transactions on Information Systems (TOIS)* 35, 4 (2017), 1–46.

[7] Alessandro Checco, Kevin Roitero, Eddy Maddalena, S Mizzaro, and G Demartini. 2017. Let's agree to disagree: Fixing agreement measures for crowdsourcing. In *Fifth AAAI Conference on Human Computation and Crowdsourcing*.

[8] Don Clark. 2012. Knowledge. http://knowledgejump.com/knowledge/knowledge.html

[9] Jared F Danker and John R Anderson. 2010. The ghosts of brain states past: remembering reactivates the brain regions engaged during encoding. *Psychological bulletin* 136, 1 (2010), 87.

[10] Ernest Davis and Gary Marcus. 2015. Commonsense reasoning and commonsense knowledge in artificial intelligence. *Commun. ACM* 58, 9 (2015), 92–103.

[11] Ujwal Gadiraju and Jie Yang. 2020. What can crowd computing do for the next generation of AI systems?. In *2020 Crowd Science Workshop: Remoteness, Fairness, and Mechanisms as Challenges of Data Supply by Humans for Automation*. CEUR, 7–13.

[12] Tianyu Gao, Xingcheng Yao, and Danqi Chen. 2021. SimCSE: Simple Contrastive Learning of Sentence Embeddings. In *EMNLP 2021*. Association for Computational Linguistics, 6894–6910.

[13] Manas Gaur, Keyur Faldu, and Amit Sheth. 2021. Semantics of the Black-Box: Can knowledge graphs help make deep learning systems more interpretable and explainable? *IEEE Internet Computing* 25, 1 (2021), 51–59.

[14] David Gundry and Sebastian Deterding. 2018. Intrinsic elicitation: A model and design approach for games collecting human subject data. In *Proceedings of the 13th International Conference on the Foundations of Digital Games*. 1–10.

[15] Amac Herdagdelen and Marco Baroni. 2010. The concept game: Better commonsense knowledge extraction by combining text mining and a game with a purpose. In *2010 AAAI Fall Symposium Series*.

[16] Aidan Hogan, Eva Blomqvist, Michael Cochez, Claudia d'Amato, Gerard De Melo, Claudio Gutierrez, Sabrina Kirrane, José Emilio Labra Gayo, Roberto Navigli, Sebastian Neumaier, et al. 2021. Knowledge graphs. *ACM Computing Surveys (CSUR)* 54, 4 (2021), 1–37.

[17] Lifu Huang, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. 2019. Cosmos QA: Machine Reading Comprehension with Contextual Commonsense Reasoning. In *Proceedings of the 2019 EMNLP-IJCNLP*. 2391–2401.

[18] Filip Ilievski, Pedro A. Szekely, Jingwei Cheng, Fu Zhang, and Ehsan Qasemi. 2020. Consolidating Commonsense Knowledge. *CoRR* abs/2006.06114 (2020). arXiv:2006.06114 https://arxiv.org/abs/2006.06114

[19] Filip Ilievski, Pedro A. Szekely, and Bin Zhang. 2021. CSKG: The CommonSense Knowledge Graph. In *The Semantic Web - 18th International Conference, ESWC 2021, Virtual Event, June 6-10, 2021, Proceedings (Lecture Notes in Computer Science, Vol. 12731)*, Ruben Verborgh and al (Eds.). Springer, 680–696.

[20] Daniel Kang, Deepti Raghavan, Peter Bailis, and Matei Zaharia. 2018. Model assertions for debugging machine learning. In *NeurIPS MLSys Workshop*.

[21] Pavan Kapanipathi, Ibrahim Abdelaziz, et al. 2021. Leveraging Abstract Meaning Representation for Knowledge Base Question Answering. *Findings of the Association for Computational Linguistics: ACL* (2021).

[22] Alicia Krebs, Alessandro Lenci, and Denis Paperno. 2018. Semeval-2018 task 10: Capturing discriminative attributes. In *Proceedings of the 12th international workshop on semantic evaluation*. 732–740.

[23] Piyawat Lertvittayakumjorn and Francesca Toni. [n. d.]. Explanation-Based Human Debugging of NLP Models: A Survey. *Framework* 3 ([n. d.]), 2.

[24] Henry Lieberman, Dustin Smith, and Alea Teeters. 2007. Common Consensus: a web-based game for collecting commonsense goals. In *ACM Workshop on Common Sense for Intelligent Interfaces*.

[25] Hugo Liu and Push Singh. 2004. ConceptNet—a practical commonsense reasoning tool-kit. *BT technology journal* 22, 4 (2004), 211–226.

[26] Gary Marcus. 2020. The next decade in ai: four steps towards robust artificial intelligence. *arXiv preprint arXiv:2002.06177* (2020).

[27] UP Narendra, BS Pradeep, and M Prabhakar. 2017. Externalization of tacit knowledge in a knowledge management system using chat bots. In *2017 3rd International Conference on Science in Information Technology (ICSITech)*. IEEE, 613–617.

[28] Ikujiro Nonaka and H Takeuchi. 1995. *The knowledge-creating company: How Japanese companies create the dynamics of innovation*. Oxford university press.

[29] Ikujirō Nonaka and Hirotaka Takeuchi. 2007. The knowledge-creating company. *Harvard business review* 85, 7/8 (2007), 162.

[30] Stefan Palan and Christian Schitter. 2018. Prolific. ac—A subject pool for online experiments. *Journal of Behavioral and Experimental Finance* 17 (2018), 22–27.

[31] Duncan Pritchard. 2013. *What is this thing called knowledge?* Routledge.

[32] Christos Rodosthenous and Loizos Michael. 2016. A hybrid approach to commonsense knowledge acquisition. In *STAIRS 2016*. IOS Press, 111–122.

[33] Wojciech Samek, Grégoire Montavon, Andrea Vedaldi, Lars Kai Hansen, and Klaus-Robert Müller. 2019. *Explainable AI: interpreting, explaining and visualizing deep learning*. Vol. 11700. Springer Nature.

[34] Wojciech Samek, Thomas Wiegand, and Klaus-Robert Müller. 2017. Explainable artificial intelligence: Understanding, visualizing and interpreting deep learning models. *arXiv preprint arXiv:1708.08296* (2017).

[35] Maarten Sap, Hannah Rashkin, Derek Chen, Ronan LeBras, and Yejin Choi. 2019. SocialIQA: Commonsense Reasoning about Social Interactions. In *Conference on Empirical Methods in Natural Language Processing*.

[36] Shaden Shaar, Firoj Alam, Giovanni Da San Martino, and Preslav Nakov. 2021. Assisting the Human Fact-Checkers: Detecting All Previously Fact-Checked Claims in a Document. *arXiv preprint arXiv:2109.07410* (2021).

[37] Elena Simperl, Maribel Acosta, and Fabian Flöck. 2013. Knowledge engineering via human computation. In *Handbook of Human Computation*. Springer, 131–151.

[38] Push Singh et al. 2002. The public acquisition of commonsense knowledge. In *Proceedings of AAAI Spring Symposium: Acquiring (and Using) Linguistic (and World) Knowledge for Information Access*.

[39] Push Singh, Thomas Lin, Erik T Mueller, Grace Lim, Travell Perkins, and Wan Li Zhu. 2002. Open mind common sense: Knowledge acquisition from the general public. In *OTM Confederated International Conferences" On the Move to Meaningful Internet Systems"*. Springer, 1223–1237.

[40] Paul Smart. 2018. Knowledge machines. *The Knowledge Engineering Review* 33 (2018).

[41] Robyn Speer, Joshua Chin, and Catherine Havasi. 2017. ConceptNet 5.5: An Open Multilingual Graph of General Knowledge. In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence*. AAAI Press, 4444–4451.

[42] Robyn Speer, Jayant Krishnamurthy, Catherine Havasi, Dustin Smith, Henry Lieberman, and Kenneth Arnold. 2009. An interface for targeted collection of common sense knowledge using a mixture model. In *Proceedings of the 14th international conference on intelligent user interfaces*. 137–146.

[43] Anselm L Strauss. 1987. *Qualitative analysis for social scientists*. Cambridge university press.

[44] Alon Talmor, Jonathan Herzig, Nicholas Lourie, and Jonathan Berant. 2019. CommonsenseQA: A Question Answering Challenge Targeting Commonsense Knowledge. In *Proceedings of the 2019 NAACL-HLT , Minneapolis, MN, USA, June 2-7, 2019, Volume 1*. Association for Computational Linguistics, 4149–4158.

[45] Luis Von Ahn. 2006. Games with a purpose. *Computer* 39, 6 (2006), 92–94.

[46] Luis Von Ahn and Laura Dabbish. 2005. ESP: Labeling Images with a Computer Game.. In *AAAI spring symposium: Knowledge collection from volunteer contributors*, Vol. 2.

[47] Luis Von Ahn, Shiry Ginosar, Mihir Kedia, Ruoran Liu, and Manuel Blum. 2006. Improving accessibility of the web with a computer game. In *Proceedings of the SIGCHI conference on Human Factors in computing systems*. 79–82.

[48] Luis Von Ahn, Mihir Kedia, and Manuel Blum. 2006. Verbosity: a game for collecting common-sense facts. In *Proceedings of the SIGCHI conference on Human Factors in computing systems*. 75–78.

[49] Luis Von Ahn, Ruoran Liu, and Manuel Blum. 2006. Peekaboom: a game for locating objects in images. In *Proceedings of the SIGCHI conference on Human Factors in computing systems*. 55–64.

[50] Liang-Jun Zang, Cong Cao, Ya-Nan Cao, Yu-Ming Wu, and CAO Cun-Gen. 2013. A survey of commonsense knowledge acquisition. *Journal of Computer Science and Technology* 28, 4 (2013), 689–719.

A APPENDIX

A.1 Additional details on our GWAP

Design choice. `FindItOut` can be adjusted to fit different requirements. Here, its parameters (*e.g.,* number of trials before a guess) were calibrated through pilot studies with crowdworkers, geared towards effectiveness and enjoyability of the game. We selected 8 and 16 cards to vary the game difficulty as players managed to formulate interesting questions with less or more effort, while still finding the game enjoyable. The relation-based templates we used to formulate questions are shown in Table 3. For SimCSE, relevant literature [12, 36] adopted top-k ($k = 3, 5, 10$) and filtered out low similarity candidates. We set $k = 5$ and similarity threshold to 0.5 for the trade-off between annotation efforts and evaluation quality.

Table 3: List of relations used in `FindItOut`.

Relation	Explicit question
IsA	Is your card a(n) _____?
HasA	Does your card have a(n) _____?
HasProperty	Is your card _____(property)?
UsedFor	Can your card be used for _____?
CapableOf	Can your card _____?
MadeOf	Is your card made of _____?
PartOf	Is your card part of (a) _____?
AtLocation	Can your card be found at _____?

Implementation. `FindItOut`'s backend API manages the game logic, and the frontend renders the game screens. The communication between the two ends consists of classic HTTP REST API for user information, JWT authentication and WebSocket for game lobbying and gameplay, allowing for continuous and bidirectional data flow between the server and client. It is written in Python and served with Flask owing to its simplicity and fast setup. All game data are stored in a PostgreSQL database. The server/client WebSocket communication is implemented using the Socket.IO library. The frontend is written using React javascript library in conjunction with Redux state library, which allows unidirectional data flow; making it predictable, easy to test and flexible.

A.2 Game Boards

In our game, the design of game boards is of great importance. To keep the game interesting, we adopted greedy search strategy to retrieve relevant concepts and generate game boards for Discriminative Attributes dataset. The algorithm to generate game boards for DA dataset can be found in Alg. 1.

Algorithm 1 The algorithm to generate DA game boards.

Require: Triple set $\mathcal{T}$, concept set C, game board size n.
Input: seed concept c_0.
Output: Game board g.
initialize game board $g = \{c_0\}$
for $i = 1 \dots n-1$ **do**
  c_i = MaximizeTripleCover(g, $C \setminus g$, $\mathcal{T}$)
  $g = g \cup c_i$
end for
return g

To generate useful knowledge for the question answering task, we based ourselves on questions of the CSQA dataset to generate game boards. Based on concepts mentioned in a question and its choices, we gather related questions and generate game boards with clustering methods, which take every question as a node and overlap of concepts between questions as edges. The algorithm to generate game boards for CSQA dataset can be found in Alg. 2.

Algorithm 2 The algorithm to generate CSQA game boards.

Require: Question-concept connection set $\mathcal{T}$, question set Q, game board size n.
Input: seed question q_0.
Output: Game board g.
initialize game board g = ObtainQuestionConcepts($\mathcal{T}$, q_0)
initialize covered question set $Q_c = \{q_0\}$
while Size(g) < n **do**
  q_i = MaximizeConceptOverlap(g, $Q \setminus Q_c$, $\mathcal{T}$)
  $g = g \cup ObtainQuestionConcepts(\mathcal{T}, q_i)$
  Q_c = FindQuestionCovered(g, Q, $\mathcal{T}$)
end while
g = FilterGameSize(g, n)
return g

A.3 Additional Results

Analysis of correctness. When tuples are incorrect, 62.3% a flipped sign, 29.9% a problematic relation, and 7.49% both a sign and a relation. Problematic relations are typically explained by a) the fact that a relation and its corresponding natural language input make sense in the question posed by the `Asker` of a round, but not necessarily in the generated tuples where the concept of the game boards might not all be related to this tuple, and b) the difficulty for some players to interpret the different relations. As for the problematic sign, it is either due to ambiguities in the meaning of a concept, or due to players forgetting to cover a card when they receive the answer to their question. Future research would be needed to optimize the post-processing to automatically identify and correct such errors, as well as to improve the user experience in order to support players in selecting the most appropriate relations, and to prompt them to cover all relevant cards at each turn.

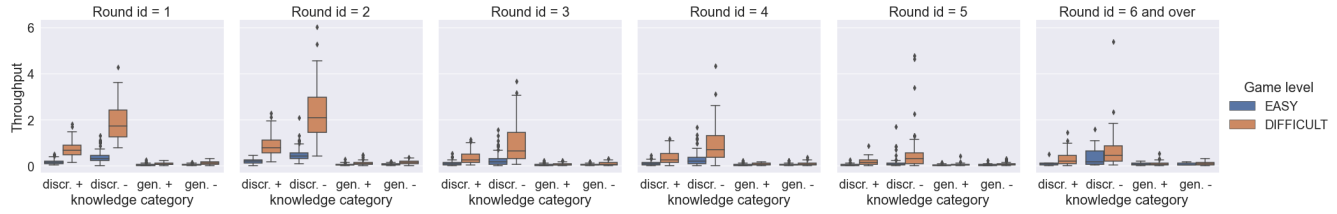
Game efficiency. Overall, 2.56% of the knowledge tuples collected within a game are overlapping, and 8.9% of the tuples collected across game boards overlap.

Table 4 present the average time taken by round across game level for both the DA and CSQA game boards. The high standard deviation for easy games in the first round is explained by the time taken by the players to learn the rules of the game.

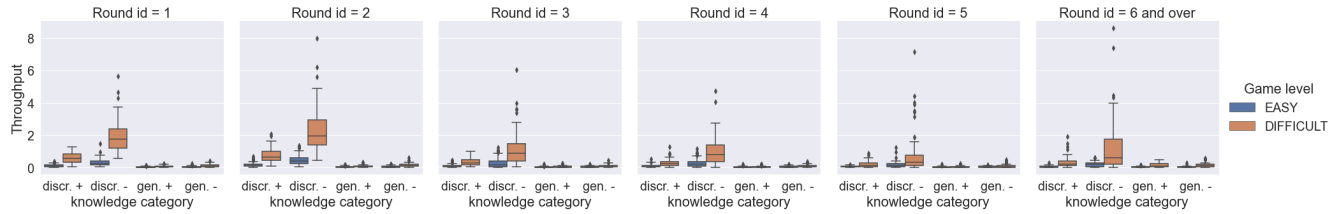
In Figure 3, we report the throughput of our game for both the DA and CSQA boards, depending on the round of the game, and the type of knowledge tuple elicited.

Table 4: Average time (in second) taken to play a round of the game (round $k = 4$ for easy games and $k = 5$ for difficult ones as more rounds are typically played for the latter).

Game board	Level	round 1	round 2	round k
DA	Easy	176.5 (*SD=735.3*)	91.9 (*SD=57.5*)	74.0 (*SD=40.8*)
	Difficult	85.8 (*SD=33.9*)	72.8 (*SD=38.7*)	66.0 (*SD=36.4*)
CSQA	Easy	141.2 (*SD=88.6*)	100.3 (*SD=68.3*)	76.8 (*SD=69.5*)
	Difficult	94.2 (*SD=46.4*)	81.7 (*SD=43.9*)	74.6 (*SD=51.3*)



(a) Discriminative attribute dataset (amount players per round: easy 72, 75, 70, 60, 40, 21 / difficult 51, 50, 49, 50, 44, 40).



(b) CSQA dataset (amount players per round: easy 70, 71, 72, 56, 33, 27 / difficult 59, 61, 55, 57, 59, 51).

Figure 3: Throughput computed over rounds of the game, and considered individually for each type of knowledge. Round 6 and over are aggregated as less players played them.

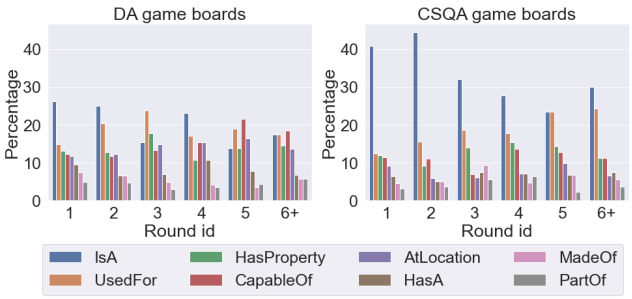


Figure 4: Relation distribution along the game rounds.

Figure 5: Player Experience Inventory questionnaire.

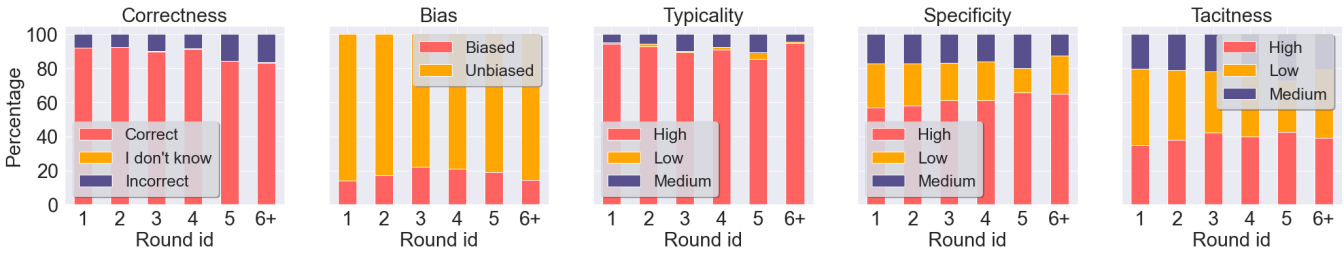


Figure 6: Bar plot illustrating the distribution of each dimension in the qualitative analysis of `FindItOut` in relation to the DA dataset, computed across different rounds.

Qualitative analysis. In Figure 6, we report the percentage of knowledge tuples falling into each of the values of our qualitative dimensions, based on the rounds of the game.

We report in Figure 4 the distribution of relations used across rounds of a game. Players tend to use explicit relations (e.g, IsA) to form the questions. After serveral rounds, tacit relations (e.g, UsedFor, PartOf) are used more often.

Enjoyability. We report in Figure 5 the enjoyability of the game. Overall, players are satisfied with the functional consequences, where the average ratings is above 1.0 (scale from -3 to 3).