

FindItOut: a Crowdsourcing Discriminative Tacit Knowledge Elicitation Multiplayer Game using Pictures

Andrea Hu¹, Agathe Balayn¹, Ujwal Gadiraju¹, Jie Yang¹

¹TU Delft

Abstract

Despite the ever-growing advances in artificial intelligence (AI), common sense acquisition and reasoning is still comparably in their early stages to other fields in AI. To further advance this field, it is necessary to collect large amounts of common sense facts or tacit knowledge to train such AI models. One effective way is to use crowdsourcing and games to make the collection process both widely available and fun at the same time. Currently, there are no tools to collect discriminative tacit knowledge efficiently and accurately. In this work, we propose FindItOut: a crowdsourcing tacit knowledge elicitation solution with multiplayer gamification and images to collect general-purpose and discriminative tacit knowledge. We show that using FindItOut, we can both efficiently and accurately collect discriminative tacit knowledge while also being fun and engaging.

Keywords— Common sense, Tacit knowledge elicitation, Crowdsourcing, GWAP, Gamification, Web-based games, ConceptNet 5, HCI

1 Introduction

Human intelligence is in large part attributed to its common sense reasoning abilities. Common sense is what we rely on to navigate concrete, everyday situations; concepts like “it is inappropriate to wear swimwear to a job interview” or “it is fine to leave a closet door open while it is not for a fridge door”, etc. AIs can be augmented with common sense to make more human-like and appropriate conversations, decisions and actions[1, 2].

Rachel Alexander[3] classifies knowledge into three categories:

Explicit Knowledge knowledge that is easy to articulate, write down, and share.

Implicit Knowledge the application of explicit knowledge. Skills that are transferable from one job to another.

Tacit Knowledge knowledge gained from personal experience that is more difficult to express.

“Common sense” can be a synonym of “tacit knowledge”, but it also usually includes implicit knowledge; these two terms will be used interchangeably in this paper. Discriminative knowledge instead is

the knowledge that allows us to differentiate one object or concept from another.

Crowdsourcing, generally speaking, is a sourcing model that collects data through a large and open group of participants often via the internet. In the context of common sense, it refers to the offloading to the crowd of computationally expensive or difficult tasks that are trivial to humans, so that they can contribute to the solution of such a task. One successful example of a crowdsourcing task is *reCAPTCHA*[4], designed originally by Luis von Ahn to digitize books and later purchased by Google. It used recognition of distorted words to distinguish between human and automated process and prevent abusive activities. It was designed to help recognise words in old books that were not clear and that could not be confidently recognised by the Optical Character Recognition (OCR).

Since the late 20th century, many common sense formal models have been built as well as different ways to elicit them[5]. The earliest models and systems were manually compiled by experts, knowledge engineers and lexicographers. Although very accurate, it was slow and required specialised personnel. In later works, crowdsourcing systems were introduced to tacit knowledge elicitation that allowed many non-specialists to enter commonsense knowledge quickly. Some of these are: *Open Mind Common Sense* (OMCS)[6] and *ConceptNet*[7]. Under this category of crowdsourced systems, there is a subcategory that specifically uses games to elicit commonsense knowledge[8]. These games are commonly known as Game With A Purpose (GWAP) as first coined by Luis von Ahn[9]. The data collected from these games are then usually used to train machine learning models.

These resources and tools mostly collected general purpose common sense knowledge. But none of these tools can directly collect discriminative knowledge. Although some tools such as ConceptNet can be used to achieve discriminative differentiation[10], it still fails on some tasks. Robyn Speer provides two such examples[11]: ‘cappuccino’ and ‘americano’ regarding ‘milk’, ‘train’ and ‘subway’ regarding ‘rails’. Currently, there are no tools specifically targeted to elicit and collect both general and discriminative knowledge.

In this work, we propose a new GWAP called FindItOut: a competitive multiplayer game that uses images to elicit general purpose and discriminative tacit knowledge. It is in big part based on ConceptNet 5[12], a revised version of the original ConceptNet. FindItOut specifically collects discriminative knowledge on closely related concepts. Although a GWAP based on images, FindItOut is designed to collect knowledge primarily on the concepts rather than the images themselves (conceptual VS perceptive). It uses images to help the users visualise the concepts, which is most useful when played on a harder difficulty.

1.1 Research Question

This research is conducted on the following research question:

“How to best design and implement a workflow of a multiplayer game for knowledge elicitation using pictures?”

Within this research question the following subquestions will also be answered:

- SQ1** How to design a game that can elicit general and discriminative knowledge?
- SQ2** What gamification elements can be added to make the game more fun and engaging?
- SQ3** How to empirically evaluate the quality (efficiency, accuracy, engagement) of the designed game?
- SQ4** What are some of the drawbacks or limitations of such a game?

The main objective is to make an engaging game that as a side effect of being played can produce valuable data about commonsense. **SQ1** and **SQ2** help design a game that is relevant and fun, while **SQ3** and **SQ4** are critical evaluations and analysis of the produced game both on the quality of the collected data and on the user engagement.

1.2 Methodology

The undertaking of this research will follow the following tasks:

1. Research and analyse existing multiplayer tacit knowledge elicitation games.
 - Type of data (tacit knowledge) collected
 - Game template and added gamification features
 - Game performance
2. Research and collect gamification techniques.
3. Design a game that can collect new tacit knowledge using multiplayer gameplay and images.
4. Implement the designed game.
5. Execute user evaluation and collect data.
6. Evaluate the game based on the collected data.

In this paper, we will show that the final designed game is able to efficiently collect accurate general and discriminative knowledge. Moreover, using a user engagement form, we demonstrate that the game is also deemed engaging and entertaining.

We first provide a description of related works in Section 2. Then a detailed description of the designed game is presented in Section 3, followed by the experimental setup in Section 4. In Section 5 we show and discuss the results of the research. Responsible research is described in Section 6. Finally, we analyse the limitations and propose future work in Section 7 and end with the conclusions in Sections 8.

2 Related Work

Over the years, many ways of collecting common sense knowledge have been devised. We describe some of the most relevant ones and their differences.

2.1 Knowledge models

Knowledge models are models capable of representing in digital form various types of knowledge that are interconnected, such as language, concepts, procedures, etc. They are computer interpretable hence enable software to interpret and make use of this knowledge. Their knowledge can be manually or automatically collected.

ConceptNet 5

As mentioned in the introduction, ConceptNet 5[12] is a revised version of the original ConceptNet[7]. It is a multilingual semantic network that captures commonsense knowledge using words (*concepts*) connected by labeled edges (*relations*). Its data is collected from multiple sources, including Open Mind Common Sense, DBpedia, Wiktionary and a couple of GWAPs.

WordNet

WordNet[13, 14] was first developed in 1985 and is an English lexical database that organises words into sets of cognitive synonyms (synset) interlinked with conceptual relations. A characteristic feature of WordNet is that it considers and distinguishes the different meanings each word might have (word sense).

2.2 GWAP

GWAPs generally come into two types: text-based and image-based, but later works also extended to include auditory, colors, etc. Text-based GWAPs collect tacit knowledge about the concepts each word represents. An example of text-based GWAP is: *Verbosity*[15]. Image-based instead can collect both conceptual and perceptive knowledge (image recognition and labelling). Most notably are: *the ESP game*[16], *Peekaboom*[17], *Phetch*[18].

Verbosity

A text-based two-player cooperative game for collecting general common sense facts with roles *Narrator* and *Guesser*. The Narrator is given a concept and tasked to give hints by filling in template sentences that are then presented to the Guesser, who has to guess the concept.

The ESP Game

An image-based two-player cooperative game for collecting labels of images. Both players are presented with the same image and must agree on the shown image by typing the same word. It also uses taboo words to restrict the possible labels and elicit more specific ones.

Peekaboom

An image-based two-player cooperative game for locating objects in images with roles *Peek* and *Boom*. Boom is shown an image and a word related to it, while Peek sees an empty board. Boom then has to reveal parts of the image to Peek so that Peek can guess the associated word. Boom can help Peek by giving hot/cold feedback to Peek's guesses or use the *ping* feature to highlight a spot on the image.

Phetch

An image-based three-to-five-player cooperative game for collecting descriptions and captions of images. There is a single *Describer* and the rest are *Seekers*. The Describer receives an image and has to type a description of the image such that the Seekers can find that image using an internal image search engine.

All the aforementioned models and GWAPs collect either general knowledge about concepts or specific knowledge about images like labels and captions. Moreover, the image-based GWAPs all use a single image each round and are not able to directly collect discriminative knowledge.

3 FindItOut

The designed game mechanic has to be able to elicit discriminative knowledge between closely related concepts. The data collection has to be both efficient and accurate. Furthermore, the game needs to be engaging and enjoyable.

3.1 Game mechanics

FindItOut is the name of the game designed to answer the research question. It is a competitive game played by two players who are matched randomly and take turns being the “Asker” and the “Replier”. At the start of the game, both players are presented with a board of multiple cards, containing pictures of various objects. The game assigns one of the cards on the board to each player as their “IT card”. The main goal of each player is to guess the opponent’s IT card by asking questions and reducing the possible candidates. The cards on the board can be flipped, which help the players keep track of the possible choices.



Fig. 1. Game screen of the asker during the asking/guessing stage of the game. Shows the asker posing a “HasProperty” type of question.

3.2 Game Flow

The game flow is described next; the diagram of which can be found in Appendix A.

1. The two players are assigned a role randomly. One being the ASKER and the other the REPLIER.
2. Start of new round. The Asker chooses an action between “ASK” and “GUESS”:
 - **ASK**: choose a question type and fill in the question, then confirm (Fig. 12). Proceed to point 3.
 - **GUESS**: choose the card that matches your guess of the Replier’s IT card on the board and confirm (Fig. 15). This action will end the game, proceed to point 6.
3. Replier receives the question and replies YES/NO according to his IT card (Fig. 16).
4. Asker receives the reply and flips the cards that don’t match the reply (Fig. 17).
5. Asker confirms and this round ends. The two players switch roles. The next round will start, go back to point 2.
6. End of game, if the guess was correct the Asker wins (Fig. 18) and the Replier loses (Fig. 19), otherwise it’s the opposite.

Using this workflow, discriminative knowledge can be collected for each game round based on the question, reply and flipping of cards. Specifically, the turning of cards tells us whether a card matches the question and answer pair, therefore discriminating between cards that were flipped and not. In Figure 1 an example of the Asker’s game screen is shown. All the game screens currently present in the game can be found in Appendix B.

3.3 Gamification techniques

To enhance the gameplay, the following gamification techniques were designed:

- Scoring system
 - Leaderboard (weekly, monthly, all time)
- Level system
 - Unlock new question types
 - Unlock harder difficulties
 - * More cards per session (16, 24, 32)
 - * Objects are more similar (harder to discriminate)
 - * Only a subset of question types allowed each round
- Timer (default 30 seconds)
- Input limit (default 2 words)
- Achievements
 - Win within X rounds ($X \in \{2, 4, 6, 8, 10\}$)
 - Y number of games played ($Y \in \{5, 10, 25, 50, \dots\}$)
 - Reached new level
 - Unlocked new relation/difficulty
- Winning Streak (1x, 2x, 3x, 4x)
- Daily Streak

Many of these techniques are taken from Luis von Ahn’s article “Designing games with a purpose”[19] and from previous GWAPs and adapted to fit FindItOut’s game mechanics. The article describes 3 design principles and various gamification techniques. These gamification techniques not only can increase the fun factor of games but can also encourage players into playing honestly and deter undesirable actions such as random inputs, spamming, etc.

To incentivise players to keep playing and therefore continue producing tacit knowledge competitively, the scoring system with a leaderboard is implemented[20]. Winning streaks, daily streaks also contribute to this end. On a more individual sense, level (difficulty) system and achievements are included. These provide the players with a progression system and can give a sense of accomplishment the more they play the game. The timer is added both as a difficulty measure, but also to ensure that the rounds don’t last too long and become boring for the other player waiting. When a player’s timer runs out, some points are deducted from the player total score as a penalty. Lastly, word limit and input validation are implemented to ensure correct gameplay and clean data.

Taboo words were not included because it is not valuable for the player to ask the concept of a specific card since by asking directly one of the concepts on the board, the player only gets to cut down that one option. It might also happen that the concepts presented include a general term and its sub-types (e.g. cat and names of species of cats). In this case, the user might want to ask whether the IT card is a cat, therefore this feature was deemed unsuitable.

Given the length of the project and the implementation time constraints, only a few of these designed techniques were applied; namely the difficulty setting (currently not available to users to configure) and input limit.

3.4 Collected knowledge

ConceptNet 5 provides in total 34 relations¹, but for the extent of FindItOut only a subset of 8 relations is used. These 8 relations were chosen based on these criteria:

1. does the relation apply to nouns?
2. is the relation easy to understand intuitively?

The 8 relations and their corresponding explicit questions are shown in Table 1.

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

Table 1: List of relations used in FindItOut

The knowledge is collected each turn, which is composed of three stages: ASK/GUESS, REPLY, FLIP. The actions in each stage contribute to the final collected knowledge.

Consider the following example question:

“Does your object have a handle? NO”

Relation $R = \text{“HasA”}$, Target $T = \text{“handle”}$,

Answer $A = \text{“NO (False)”}$, Concept $C = \text{object in the card}$

| From | To | Action | Knowledge | Weight |
|-------|-------|-----------|--------------------|--------|
| Uncov | Uncov | Unchanged | $R(C, T) = A$ | Medium |
| Uncov | Cov | Flipped | $R(C, T) = \neg A$ | Medium |
| Cov | Uncov | Unflipped | $R(C, T) = A$ | High |
| Cov | Cov | Unchanged | $R(C, T) = False$ | Low |

Table 2: Collected knowledge based on player actions.
 $R = \text{Relation}$, $T = \text{Target}$, $C = \text{Concept}$, $A = \text{Answer}$

Based on the Replier’s answer, the relation’s knowledge is negated. The weight describes how reliable the notion is. In the case when a previously covered card is flipped around again, the weight is high because this action is considered highly intentional. Instead, the weight is low when a covered card is kept covered, as players don’t often flip back already covered cards. In this case, the knowledge result is always set as False. This might lead to false negatives and should be taken into account when using the low weight assertions.

This collected knowledge is both general, but also discriminative since covering some cards while leaving others unchanged discriminates the knowledge between the two types of cards.

Furthermore, it is also possible to collect typicality knowledge. This is especially true towards the end of the game when the cards that are still uncovered are most probably very similar (imagine a bass clarinet and a bassoon).

At the end of a game session, the rounds data is turned into the following tacit knowledge format:

¹<https://github.com/commonsense/conceptnet5/wiki/Relations>

Relation type One of the 8 relations shown in Table 1.

Target The free text input inserted by the Asker (max 2 words).

Object The object the relation applies to (the object in the card).

Result 1 if relation applies positively or -1 if negatively (or True/False). Shown in Table 2.

Weight Confidence of this assertion, can be one of the following values:

- **I:** Invalid
- **L:** Low
- **M:** Medium
- **H:** High

3.5 Cheating

As with most games, there will be players who try various ways to cheat or disrupt the correct execution of the game. As for FindItOut, the aspect that is most vulnerable to this kind of behaviour is the free text input when the Asker asks. In this case, it is not in the best interest of the user to insert random or irrelevant information as it will prevent the player to progress the game. But to further prevent irregular actions, the inserted words are limited to 2 words and can be matched with a dictionary to ensure that they are English words or invalidate offensive words.

On the other hand for the Replier, it is in his best interest to lie when replying to a proposed question, such that the Asker will be misled. Fortunately, the game design allows to identify and account for this.

Keeping in mind the Replier’s *IT* card and the question (relation R), after the Asker ends turning the cards, the following cases can happen:

- If the Replier’s *IT* card is still uncovered on the Asker’s board \Rightarrow the collected knowledge is valid
- If the Replier’s *IT* card is covered by the Asker \Rightarrow there are two possibilities:
 - Replier did not reply truthfully (this relation can be ignored).
 - Asker made a mistake (this relation can be ignored).

Therefore, when the opponent’s *IT card* is flipped by the Asker, that round’s data can be seen as invalid and ignored.

In general, the game was designed to be very restrictive: not allowing players to perform any actions while not their turn, limiting the word input to 2 words and preventing special characters. This contributes to the quality of the collected data.

In future updates, the history of questions asked can be shown to the players at the end of a game and report the opponent if he did not reply truthfully. These game sessions can then be inspected by the game team and invalidated if not accurate.

4 Experimental Setup

The assessment of the game will happen on two ends: the collected data and user engagement. The data is analysed quantitatively and qualitatively on efficiency and accuracy. To measure efficiency we will determine the throughput, which is the number of problem instances solved per human minute (or hour). A problem instance in FindItOut is one assertion collected from a question and answer pair and the flipping of a single card (e.g. if the board has 16 cards, 16 assertions will be made each round). Regarding accuracy, we will randomly sample 10 assertions for each question type and manually evaluate their quality and correctness. Lastly, the user engagement will be evaluated according to the user engagement scale short form[21]. In this section, we describe the details of the implementation and game generation algorithms.

4.1 Implementation

FindItOut is implemented as a web app (or web game) and can be played at <https://finditout.vercel.app>. The web app platform ensures convenience and portability: the game can be served on any platform as long as it has a browser. It is implemented to be responsive, such that it can adapt to any screen size whether it is mobile or desktop.

The general structure of FindItOut is divided into two components: backend API for managing most of the game logic and the frontend that renders the game screens. The communication between the two ends consists of two methods: classic HTTP REST API for user information, JWT authentication and WebSocket for game lobbying and gameplay. The use of WebSockets allows for continuous and bidirectional data flow between server and client, perfect for real-time gameplay.

The backend server is written in Python and served with Flask for its simplicity and fast setup. It provides access points for JWT authentication. All the game states are stored in the server, so the connections between the two players always run through the server. This allows the players to stay in the game even in case of a network interruption. The authentication data, game sessions and collected knowledge are persisted in a PostgreSQL database, which only the server accesses. The passwords are hashed and kept safe as to current security standards. 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; therefore predictable, easy to test and flexible.

4.2 Game session generation

FindItOut in great part builds upon the works of ConceptNet 5. Specifically, it uses its relationships to find related words. The variables of a game session are: difficulty $d \in [0, 1]$, the number of cards $n \in \{16, 24, 32\}$. The generation proceeds as follows:

1. Generate a set of related words (Algorithm 1)
2. Choose n cards based on the game's difficulty. (Algorithm 2)
3. Collect images for each object in the set using Google Image.
4. Choose 2 distinct objects as IT cards for players 1 and 2.
5. Start game

Algorithm 1: Generation of set of related words

Input: $n = \text{num cards}$
Result: Set of related words R

```
1  $R \leftarrow \text{set}()$ 
2  $s \leftarrow$  Choose one random seed word from a
  predetermined list of 200 picturable words[22]
3  $R.add(s)$ 
4 while  $size(R) < n$  do
5    $w \leftarrow$  random word from  $R$ 
6    $T \leftarrow$  retrieve a list of 100 related words to  $w$  using
    ConceptNet 5's API
7    $T \leftarrow$  filter  $T$  by Noun using WordNet pos
8    $T \leftarrow$  filter by concreteness using empirical dataset
    of word concreteness[23]
9    $R.update(T)$ 
10 end
```

The generation of a game starts from a seed word that is taken from a list of 200 picturable words[22]. Then using ConceptNet

5.8's related concepts API, a list of related concepts is parsed. These concepts come with a weight value in the range of $[0, 1]$; it represents how confident ConceptNet 5 is about the validity of relatedness between the seed word and the new word. In other words, how many times an assertion with the two concepts have appeared together from different sources. The later Algorithm 2 is sorted based on this weight measure. Then two stages of filtering are applied to make sure that the concepts are also picturable nouns. The algorithm is repeated if not enough concepts are left after the filtering. In the case this happens, the weights of the new concepts are multiplied by the chosen seed word's weight.

Algorithm 2: Choose n from ordered list with gaussian weights

Input: $d = \text{difficulty}$, $R = \text{elements}$
Result: List of n elements from a list R

```
1  $size \leftarrow size(R)$ 
2  $\mu \leftarrow size * (1 - clamp(0, d, 1))$ 
3  $\sigma \leftarrow size / F$  ( $F = 3$  default)
4  $norm \leftarrow \text{normal\_distribution}(\text{center}=\mu, \text{var}=\sigma)$ 
5  $W \leftarrow \text{norm.pdf}(n)$  for  $n = 0 \rightarrow size$ 
6  $W \leftarrow \text{normalize } W$ 
```

Once a set of related concepts are collected, a subset of n cards has to be selected for the game session. The variable n is determined by the difficulty of the game. The difficulty variable is defined as a decimal number within $[0, 1]$, with 0 being the easiest setting and 1 the hardest. The input list is sorted with decreasing weight (relatedness confidence). The weighted choice of items is based on a gaussian distribution with varying centre. The Gaussian distribution has two variables: centre and variance. The centre is the size of the input array multiplied by $1 - \text{difficulty}$, meaning the harder the difficulty ($d \rightarrow 1$) the more closely related the terms ($w \rightarrow 0$). The variance is calculated by dividing the size by a difficulty range factor $F > 0$. As F increases, the resulting normal distribution's width decreases (narrows). The default value of 3 is chosen to make 3σ of one side fit the whole list. In the end, a list of n items is returned.

For the next step, the images of each concept are collected using Google Image Search API. The search is performed directly on the concept's name and is filtered with transparent background and png file type. For each concept, 5 images' links are stored in the database for caching and faster retrieval in future. For every game, one of the 5 images is chosen uniformly randomly as the object's image.

The last step before starting the game is to choose two objects for the players. This choice is a uniform exclusive choice, so the players will not have the same object. By using an exclusive choice, the collected knowledge can be more diverse per game. From preliminary internal testing, it was found that players tended to ask the same questions. Thus, having different cards would help diverge the answers even if the same questions were asked.

For the actual experiment, some variables that would be otherwise adjustable by the player were fixed because of implementation time restraints. The variables used for the game generation are 16 cards per game and 0.2 for difficulty (centre of normal distribution).

5 Results and Discussion

5.1 Efficiency

The data collection lasted for five days for 36 game sessions. From these 36 game sessions, a total of 3408 assertions were collected. An overview of the usage of relation types in these assertions is shown

in Figure 2. The invalid values represent all the knowledge points that were invalidated for the reasons mentioned in Section 3.5.

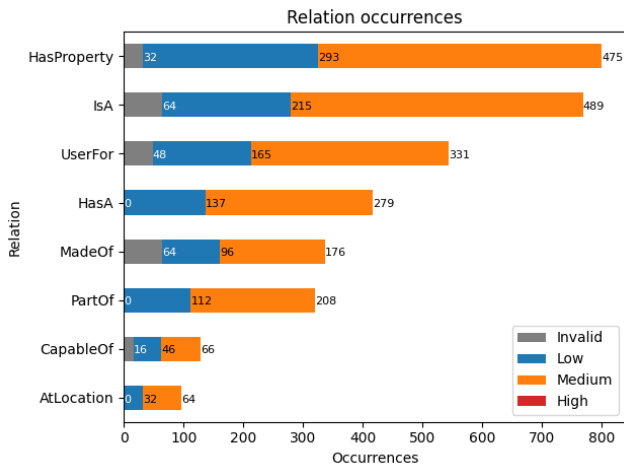


Fig. 2. Occurrences of each relation by weights.

The two most used relations are HasProperty and IsA. This might be because these two relation types apply best to objects and are the most general. Also, when the objects are similar, they often share some properties such as use, location, material, etc. Therefore it is harder to use those relations to distinguish the objects.

Of all the assertions, 6.6% of them were marked invalid. Analysing the invalidated data, most of them were because of interpretation mistakes, meaning the Replier answered truthfully, but either the two players have perceived the question differently or because the question could not be answered with a straightforward yes or no (ambiguous or it depended). An example of such a question with “mattress” as Replier’s *IT card* is: “Can your object be used for sitting? Yes”. The Asker flipped the mattress card, most probably because the primary use of mattresses is to lie down and not for sitting (some of the other cards were chair related).

The efficiency is calculated as the amount of problems instances solved per human minute. From the 36 game sessions, FindItOut was able to obtain on average 11.17 (std. dev. = 4.22) assertions per minute with two players, including invalidated assertions (Figure 3). Excluding the invalidated data, the game achieved an efficiency of 10.45 (std. dev. = 4.36) assertions per minute (Figure 4). In comparison, The ESP game was able to collect on average 3.89 (std. dev. = 0.69) labels per minute with two players[16].

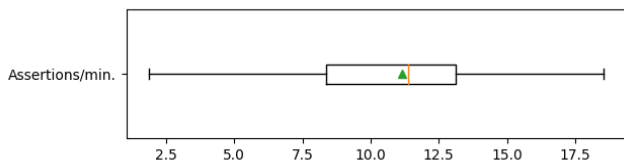


Fig. 3. Boxplot of assertions per minute with two players including invalidated data.

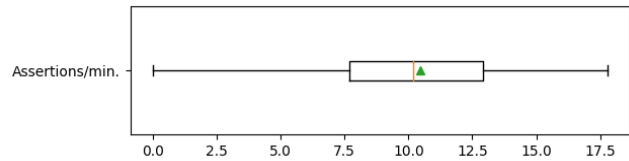


Fig. 4. Boxplot of assertions per minute with two players excluding invalidated data.

The standard deviation is moderately high mostly because of two reasons: the timer was not yet implemented to urge players and regulate the time, but especially the players had to spend time looking up the concepts they were unfamiliar with and dragged the game time on. This leads to game sessions that usually ended up lasting for quite a long time, but in cases where the concepts were relatively well known, the game was able to collect at the highest 20.9 assertions per minute. If the timer and the word dictionary are implemented, this figure is expected to increase, also as the players get more familiar with the game (learning factor).

5.2 Accuracy

To evaluate the accuracy of the collected knowledge we randomly sampled 10 assertions from each relation type and checked whether they were valid and made sense. The full list can be found in Appendix D. The table contains a column marked as “Incorrect” and can have three values: empty if correct, ‘/’ if it is debatable and ‘X’ if clearly incorrect. We can confidently say that almost all assertions are valid; specifically, out of 80 sampled questions, 4 were ‘/’ and 4 were ‘X’. Some of the *Low* weight assertions are not accurate, most probably because they were turned in earlier rounds and the player chose not to flip them back even if the question applied to that object. This behaviour is expected and should be accounted for when using the collected data. As for the clearly incorrect, most often it happened because of two possible reasons: the image was an incorrect representation of the concept so mislead the Asker or the Asker did not know the concept clearly. As an example of the former case, the assertion ⟨PartOf, auricula, ear, False, M) was incorrect because the image was a heart but auricula is the outer ear. An example of the latter case is the assertion ⟨CapableOf, pintle, screwing something, True, M) where the Asker might have mistaken a pintle for a tool to screw. If played by enough people, these mistakes and outliers will be mitigated under the assumption that the images correctly represent the concept.

5.3 Player engagement

To evaluate the player engagement, an anonymous 12-question user engagement scale short form[21] was given to the players to fill out after experiencing the game. The questionnaire also included at the end one optional question asking for comments and recommendations, which provided insight into the questionnaire ratings. At the end of the experiment, 8 players have filled out the form. The final results are shown in Table 3 and the detailed results can be found in Appendix C.

| Subscale | Average score | Percentage |
|------------------------|---------------|------------|
| FA Focused Attention | 3.79/5 | 75.8% |
| PU Perceived Usability | 2.63/5 | 52.5% |
| AE Aesthetic Appeal | 3.58/5 | 71.7% |
| RW Reward | 4.29/5 | 85.8% |
| Final engagement | 3.57/5 | 71.5% |

Table 3: Results of the user engagement scale short form

The lowest scoring subscale is Perceived Usability, specifically the PU-S.1 and PU-S.3 questions. These questions are “I felt frustrated while using FindItOut ” and “Using FindItOut was taxing”. The replies to the last optional question showed that players found having to look up the lesser-known terms very cumbersome and broke the immersion. It was also mentioned that sometimes the images would not be representative of the concepts, therefore confusing or misleading. Comparingly, the Reward RW subscale is scored highest. The players responded that by using FindItOut they learned many new terms and that winning was very satisfying.

Another behavioural measure is the replay value. 16 times out of the 36 (44%) game sessions the players played again. A possible contributing factor of this figure is the near-win[24] or near-miss[25] effect. As the game progresses, each player is left with fewer and fewer cards open on the board. When the game ends, the loser most often will be left with a single or just a few choices left. This leads to a feeling of almost winning, which in turn increases the desire to continue playing a new game or a rematch[25].

6 Responsible Research

When collecting data from crowds, it is necessary to consider the type of data that is being collected. FindItOut collects and stores authorization credentials with encrypted password, their inputted tacit knowledge and gameplay-related data such as playtime, scores, etc. According to the GDPR, only the credentials can be regarded as indirectly personally identifiable information. To ensure that this information is kept private, all connections with the users and their credentials will be removed at the end of the experiment when the tacit knowledge is compiled. The game infrastructure is also implemented following cybersecurity standards to minimize the possibility of data breaches and attacks.

Although not open-source, the game flow, game rules, gamification techniques and game generation algorithms of FindItOut are clearly described in Section 3, therefore the game and experiment are easily reproducible. Throughout the game pipeline, multiple random values are used, but they don’t have a huge impact on the quality of data collected since they are used for generating a random game session. If played by enough people, most concepts will be covered.

FindItOut makes use of various APIs, like ConceptNet 5, Wordnet and Google Image Search. These are all public APIs and therefore available to everybody. Regarding Google Image Search, the images are never downloaded nor stored on the server or locally, instead the original link of the image resource is saved and sent to the players. This ensures that the images are correctly attributed to the source.

It is worth noting that what FindItOut achieves is only limited to elicitation and collection of tacit knowledge, but in no part trains machine learning models or uses AI. It is though to some extent limited by the resources it uses, namely ConceptNet 5, Wordnet pos, word concreteness and Google Image Search.

7 Limitations and Future Work

The game design of FindItOut can be applied to other languages as well and not just limited to English. Since ConceptNet 5 is a multilingual semantic network, the game generation can be easily changed to another language just by changing the API query language setting. It is necessary though to update the other stages of the game generation, such as the seed word list and concreteness filtering (which can be skipped if concrete nouns are not required). As for WordNet, although originally an English-only resource, it can be extended with Multilingual Wordnet[26], which includes over 100 languages.

Implementation

The weakest link in the chain of FindItOut’s game generation is the image parsing stage. Currently, it uses Google Image Search (GIS) to parse the first few most relevant results, but sometimes the obtained image is either not an accurate representation of the concept or is an invalid resource (not displayable). Originally the choice of using GIS was made because there is currently no publicly available dataset of clean images (transparent background) of objects. This problem can be fixed if such a dataset is created or made publicly available.

Furthermore, the issue of image parsing is not just this simple. FindItOut faces many of the challenges that natural language processing (NLP) faces. For example, words can have multiple meanings. Take “nut” as an example, it can be both a fruit and a hardware (nut and bolt). Since FindItOut works with single concepts and not sentences or corpus, there is no contextual information to distinguish one meaning from the other. One possible direction is to use Wordnet’s word sense to distinguish the specific sense. But this would then require being able to search images depending on the word sense, which can be on its own rather challenging. Moreover, if word senses were to be implemented, then the relatedness measure used during the game generation can also be substituted by WordNet’s similarity feature, of which WordNet provides three types[27]. These measures can be a better representation of concept similarity over ConceptNet 5’s confidence weight.

A possible solution to the two above problems is the hint feature. It allows the Asker to automatically turn around a single wrong card at the cost of a bonus task. This feature should be limited to a single use in a game. To get access to this hint, the player has to complete an image labelling task: the player is shown an image and a concept (plus its definition if word sense is taken into account) and has to choose whether this image is an appropriate representation of the concept. Through this feature, FindItOut can slowly build up a dataset of images of concepts and over time overcome the aforementioned problems.

Another limitation of FindItOut is the list of seed words, which currently amounts to 200 words. Despite the expansion of concepts using ConceptNet 5, this expansion is still within the circle of related concepts to these 200 words. This means that only a subset of all possible concrete nouns/objects is available to the game. To overcome this, the game could add new words to the seed list based on a metric. For example words with enough assertions and that the game is confident in its concreteness (e.g. without many contradicting assertions). This would need further testing.

On a more general note, the players can exhibit a sort of “anchoring effect”, where the produced tacit knowledge is tightly related to the specific image shown to the player and not the general concept itself; an example of this is the colour of the objects. The current implemented solution to this is to randomise the shown image for every object. If applied over a large number of players this effect will be mitigated. Besides, at the start of a game, the

players are shown a disclaimer saying that the images might not be an accurate representation of the concepts, therefore the questions made should be primarily based on the concepts and only if necessary based on the images.

To improve user experience, some ‘Quality-of-Life’ features can also be implemented. We list three of such features.

1. A dictionary definition can be added to each card. When a player clicks or hovers on a card, its concept’s definitions are shown. This can greatly reduce the game time since, during the evaluation, the players spent a considerable amount of time looking up the lesser-known words. Not only it can reduce game time and therefore perceived usability, but also increase the reward of playing FindItOut as players can both have fun and learn new words.
2. In addition, a question and reply history can be provided to players to help them keep track of the questions already asked. This history can also be shown at the end of the game to let players confirm that the opponent replied truthfully, otherwise they can report the other player.
3. At the end of a game, if the Asker wins, the Replier can have a chance to still make a guess. If he guesses correctly then he will also be rewarded a smaller amount of points, or else no points are given. This can have adverse effects on the engagement, as it might reduce the near-win effect. This would need further experimenting.

A possible problem that players have brought up is that sometimes the concepts are synonyms, thus too similar and hard to discriminate. To solve this, synonyms within a board can be first grouped and seen as one concept. If this synonym group is chosen during the game generation, then a random synonym within that group is selected. To find these group of synonyms we can use the synsets available in WordNet, described in Section 2.1.

To reduce the possibility of invalid, unclear or ambiguous questions, a “NOT CLEAR” answer can be added as an option for the Replier. In this way, if the Replier chooses NOT CLEAR, the Asker has to ask a new question again or rephrase the previous question. This feature is shown in Appendix A. It is worth noting that a NOT CLEAR answer can also provide the Asker with valuable insight about the Replier’s IT card. This is because it should be very straightforward to give a yes/no answer for most of the objects, but a NOT CLEAR answer can let the Asker know that the opponent’s IT card can be ambiguous regarding that question and greatly reduce the options. This feature would need extensive testing to determine if valuable and if it can be exploited.

Evaluation

Due to the short time frame of the project, we were not able to collect many game sessions and questionnaire replies. Therefore, the results are not very conclusive but do give a rough idea of the quality and performance of the game.

In regards to the efficiency, Luis von Ahn proposes another measure: the expected contribution[19]. It is calculated as throughput multiplied by average lifetime play(ALP). Throughput is the number of problem instances solved per human hour (similar to the evaluation measure used in this work), while ALP is the average time a player plays the game overall. Because of the lack of evaluation data and high variance, this measure was deemed unsuitable for this experiment.

For accuracy instead, since the collected knowledge is generative, there are no ground truths available to test the accuracy on. A possible future direction is to either check a more indicative amount of samples or devise a discriminative knowledge test suite.

8 Conclusion

In this work we have presented FindItOut, a two-player competitive game to elicit general and discriminative knowledge. The collected discriminative knowledge can allow software and AI to distinguish between similar concepts. We have shown in the evaluation that the game is both fun to play and capable of efficiently and accurately collecting discriminative knowledge. Even though the evaluation period lasted only for less than a week with a dozen players, we were able to collect over 3000 assertions. We have also thoroughly discussed multiple ways to improve the current game architecture. FindItOut shows considerable potential, both as an engaging game and as a tacit knowledge eliciting tool.

Common sense knowledge (and its elicitation) is becoming increasingly important as AI approaches in achieving human-like conversations, decisions and actions. We believe that FindItOut can largely contribute to this end.

9 Acknowledgements

We would like to thank Sharon Lim Yu Jung, Ting Xin, Philip Huang for their unconditional support throughout the whole process. We also thank all the players who contributed to the results of this work.

Soli Deo Gloria

References

- [1] Forough Arabshahi et al. “Conversational Neuro-Symbolic Commonsense Reasoning”. In: *arXiv:2006.10022 [cs, stat]* (Feb. 2021). arXiv: 2006.10022. URL: <http://arxiv.org/abs/2006.10022>.
- [2] Kaixin Ma et al. *Knowledge-driven Data Construction for Zero-shot Evaluation in Commonsense Question Answering*. Dec. 2020.
- [3] *Different Types of Knowledge: Implicit, Tacit, and Explicit*. en-US. Jan. 2018. URL: <https://bloomfire.com/blog/implicit-tacit-explicit-knowledge/>.
- [4] Luis Ahn et al. “reCAPTCHA: Human-based character recognition via Web security measures”. In: *Science (New York, N.Y.)* 321 (Sept. 2008), pp. 1465–8. DOI: 10.1126/science.1160379.
- [5] Erik T. Mueller. “Chapter 19 - Acquisition of Commonsense Knowledge”. en. In: *Commonsense Reasoning (Second Edition)*. Ed. by Erik T. Mueller. Boston: Morgan Kaufmann, Jan. 2015, pp. 339–363. DOI: 10.1016/B978-0-12-801416-5.00019-X. URL: <https://www.sciencedirect.com/science/article/pii/B978012801416500019X>.
- [6] Push Singh et al. “Open Mind Common Sense: Knowledge Acquisition from the General Public”. en. In: *On the Move to Meaningful Internet Systems 2002: CoopIS, DOA, and ODBASE*. Ed. by Robert Meersman and Zahir Tari. Lecture Notes in Computer Science. Berlin, Heidelberg: Springer, 2002, pp. 1223–1237. DOI: 10.1007/3-540-36124-3_77.

- [7] H. Liu and Push Singh. “ConceptNet — A Practical Commonsense Reasoning Tool-Kit”. In: (2004). DOI: 10.1023/B:BTTJ.0000047600.45421.6D.
- [8] John Chamberlain et al. *Using Games to Create Language Resources: Successes and Limitations of the Approach*. en. Pages: 42. Springer, Jan. 2013. URL: <https://hal-lirmm.ccsd.cnrs.fr/lirmm-00831442>.
- [9] L. von Ahn. “Games with a purpose”. In: *Computer* 39.6 (June 2006). Conference Name: Computer, pp. 92–94. DOI: 10.1109/MC.2006.196.
- [10] Robyn Speer and Joanna Lowry-Duda. “Luminoso at SemEval-2018 Task 10: Distinguishing Attributes Using Text Corpora and Relational Knowledge”. In: *Proceedings of The 12th International Workshop on Semantic Evaluation*. New Orleans, Louisiana: Association for Computational Linguistics, June 2018, pp. 985–989. DOI: 10.18653/v1/S18-1162. URL: <https://www.aclweb.org/anthology/S18-1162>.
- [11] Robyn Speer. *Tutorial: Distinguishing attributes using ConceptNet*. en. Sept. 2018. URL: <http://blog.conceptnet.io/posts/2018/distinguishing-attributes-using-conceptnet/>.
- [12] Robyn Speer, Joshua Chin, and Catherine Havasi. “ConceptNet 5.5: An Open Multilingual Graph of General Knowledge”. en. In: (2017), pp. 4444–4451. URL: <http://aaai.org/ocs/index.php/AAAI/AAAI17/paper/view/14972>.
- [13] George A. Miller. “WordNet: a lexical database for English”. In: *Communications of the ACM* 38.11 (Nov. 1995), pp. 39–41. DOI: 10.1145/219717.219748. URL: <https://doi.org/10.1145/219717.219748>.
- [14] Christiane Fellbaum. *WordNet: An Electronic Lexical Database*. en. Google-Books-ID: Rehu8OOzMIMC. MIT Press, 1998.
- [15] Luis von Ahn, Mihir Kedia, and Manuel Blum. “Verbosity: a game for collecting common-sense facts”. In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. CHI ’06. New York, NY, USA: Association for Computing Machinery, Apr. 2006, pp. 75–78. DOI: 10.1145/1124772.1124784. URL: <https://doi.org/10.1145/1124772.1124784>.
- [16] Luis von Ahn and Laura Dabbish. “Labeling images with a computer game”. In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. CHI ’04. New York, NY, USA: Association for Computing Machinery, Apr. 2004, pp. 319–326. DOI: 10.1145/985692.985733. URL: <https://doi.org/10.1145/985692.985733>.
- [17] Luis von Ahn, Ruoran Liu, and Manuel Blum. “Peekaboom: a game for locating objects in images”. In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. CHI ’06. New York, NY, USA: Association for Computing Machinery, Apr. 2006, pp. 55–64. DOI: 10.1145/1124772.1124782. URL: <https://doi.org/10.1145/1124772.1124782>.
- [18] Luis von Ahn et al. “Improving Image Search with PHETCH”. In: *2007 IEEE International Conference on Acoustics, Speech and Signal Processing - ICASSP ’07*. Vol. 4. ISSN: 2379-190X. Apr. 2007, pp. IV-1209–IV-1212. DOI: 10.1109/ICASSP.2007.367293.
- [19] Luis von Ahn and Laura Dabbish. “Designing games with a purpose”. In: *Communications of the ACM* 51.8 (Aug. 2008), pp. 58–67. DOI: 10.1145/1378704.1378719. URL: <https://doi.org/10.1145/1378704.1378719>.
- [20] Biyun Huang and Khe Hew. “Do points, badges and leaderboard increase learning and activity: A quasi-experiment on the effects of gamification”. In: Dec. 2015.
- [21] Heather L. O’Brien, Paul Cairns, and Mark Hall. “A practical approach to measuring user engagement with the refined user engagement scale (UES) and new UES short form”. en. In: *International Journal of Human-Computer Studies* 112 (Apr. 2018), pp. 28–39. DOI: 10.1016/j.ijhcs.2018.01.004. URL: <https://www.sciencedirect.com/science/article/pii/S1071581918300041>.
- [22] *Picturable Words*. URL: <http://ogden.basic-english.org/wordpic0.html>.
- [23] Marc Brysbaert, Amy Beth Warriner, and Victor Kuperman. “Concreteness ratings for 40 thousand generally known English word lemmas”. en. In: *Behavior Research Methods* 46.3 (Sept. 2014), pp. 904–911. DOI: 10.3758/s13428-013-0403-5. URL: <https://doi.org/10.3758/s13428-013-0403-5>.
- [24] Monica Wadhwa and JeeHye Christine Kim. “Can a Near Win Kindle Motivation? The Impact of Nearly Winning on Motivation for Unrelated Rewards”. en. In: *Psychological Science* 26.6 (June 2015). Publisher: SAGE Publications Inc, pp. 701–708. DOI: 10.1177/0956797614568681. URL: <https://doi.org/10.1177/0956797614568681>.
- [25] Luke Clark et al. “Gambling Near-Misses Enhance Motivation to Gamble and Recruit Win-Related Brain Circuitry”. en. In: *Neuron* 61.3 (Feb. 2009), pp. 481–490. DOI: 10.1016/j.neuron.2008.12.031. URL: <https://www.sciencedirect.com/science/article/pii/S0896627309000373>.
- [26] Francis Bond and Ryan Foster. “Linking and Extending an Open Multilingual Wordnet”. en. In: (), p. 11.
- [27] Ted Pedersen, Siddharth Patwardhan, and Jason Michelizzi. “WordNet::Similarity: measuring the relatedness of concepts”. en. In: *Demonstration Papers at HLT-NAACL 2004 on XX - HLT-NAACL ’04*. Boston, Massachusetts: Association for Computational Linguistics, 2004, pp. 38–41. DOI: 10.3115/1614025.1614037. URL: <http://portal.acm.org/citation.cfm?doid=1614025.1614037>.

A Game flow

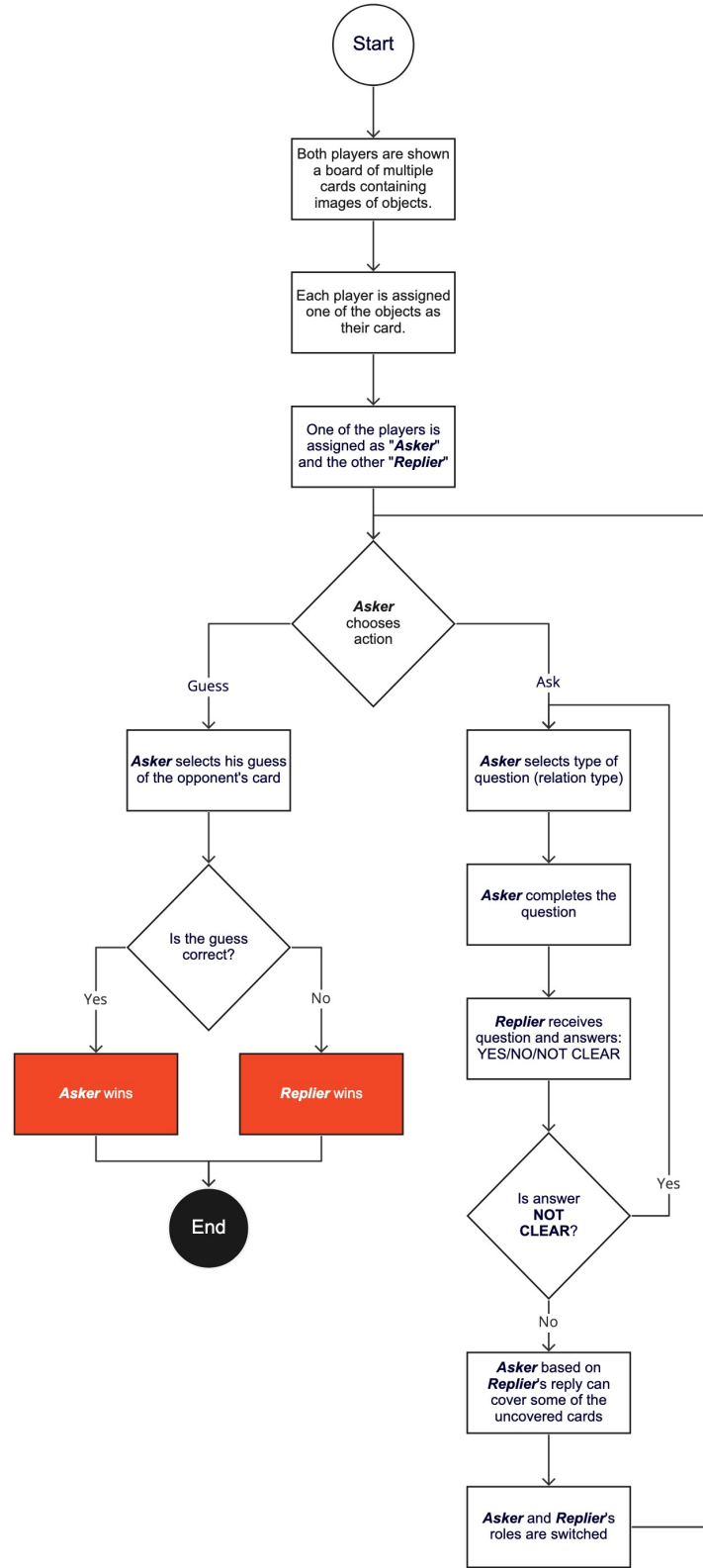


Fig. 5. Game flow of FindItOut.

B Game screens

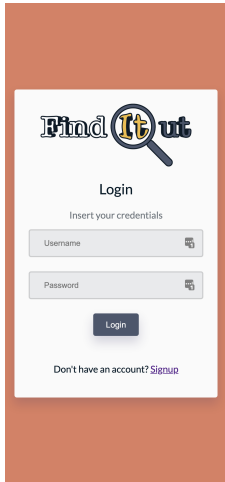


Fig. 6. Login page

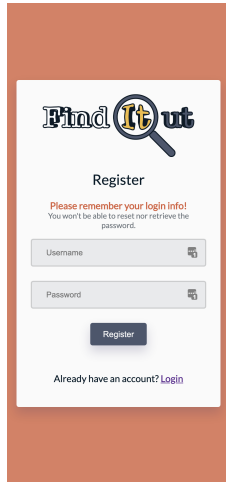


Fig. 7. Register page

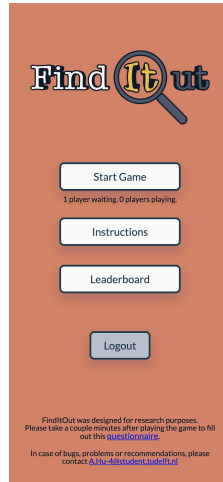


Fig. 8. Main menu

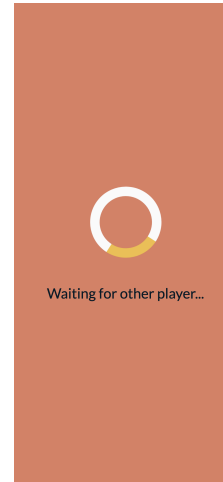


Fig. 9. Lobby page

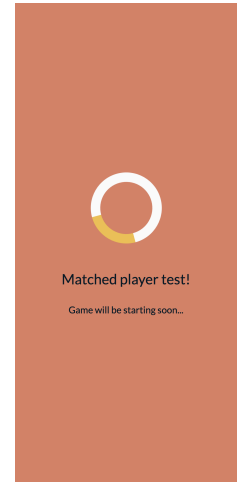


Fig. 10. Game starting

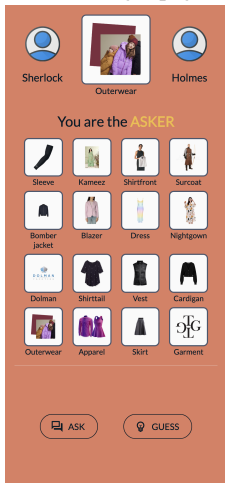


Fig. 11. Initial game screen

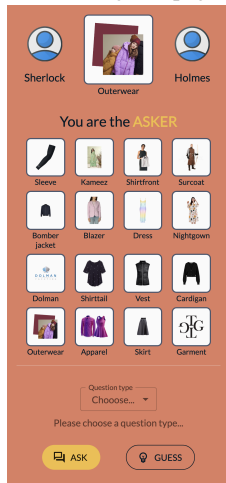


Fig. 12. Asker asking

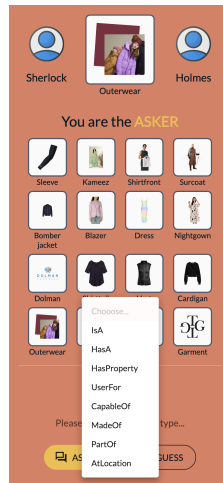


Fig. 13. Asker full question type

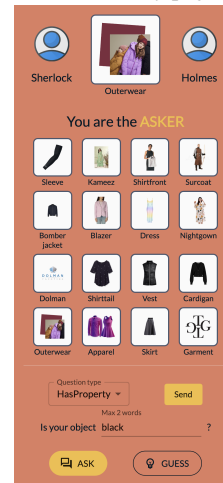


Fig. 14. Asker full question

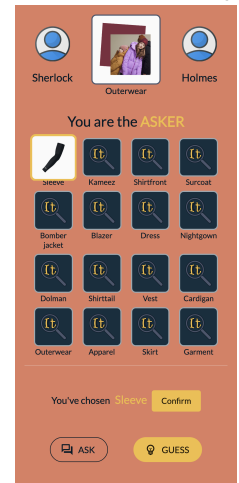


Fig. 15. Asker guesses

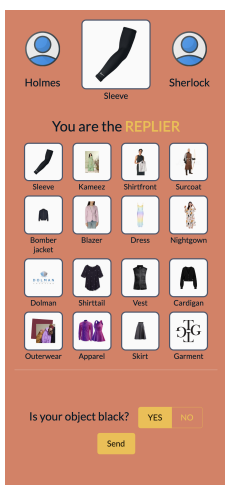


Fig. 16. Replier replies

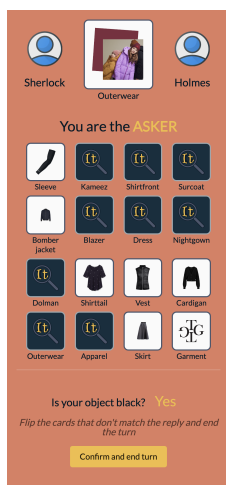


Fig. 17. Asker flips

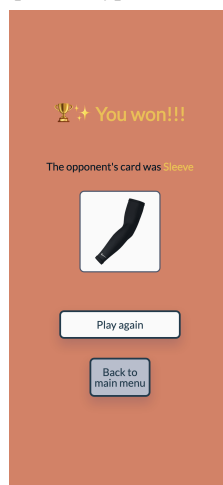


Fig. 18. Win screen

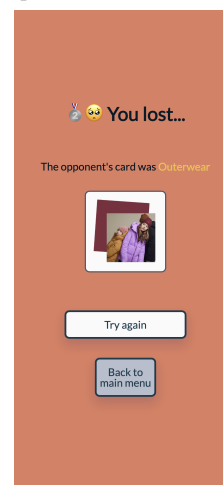


Fig. 19. Lose screen

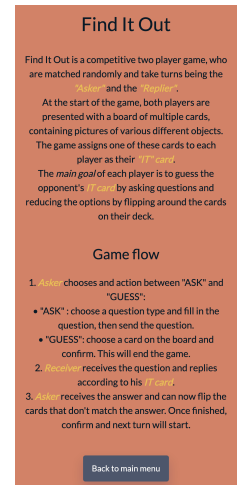


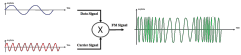





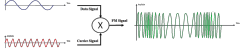













Fig. 20. Instructions page



















C User evaluation form
















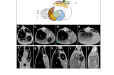




| # | UES dimension identifier | Question | Strongly disagree ... Strongly agree | Avg. | Var. | Dev. | Med. |
|----|--------------------------|--|--------------------------------------|------|------|------|------|
| 1 | FA-S.1 | I lost myself in this experience | | 3.8 | 1.1 | 1.0 | 4 |
| 2 | FA-S.2 | The time I spent using FindItOut just slipped away | | 4.0 | 1.1 | 1.1 | 4 |
| 3 | FA-S.3 | I was absorbed in this experience | | 3.6 | 0.8 | 0.9 | 4 |
| 4 | PU-S.1 | I felt frustrated while using FindItOut | | 2.1 | 1.3 | 1.1 | 2 |
| 5 | PU-S.2 | I found FindItOut confusing to use | | 1.8 | 0.5 | 0.7 | 2 |
| 6 | PU-S.3 | Using FindItOut was taxing | | 3.3 | 1.4 | 1.2 | 4 |
| 7 | AE-S.1 | FindItOut was attractive | | 3.5 | 1.4 | 1.2 | 4 |
| 8 | AE-S.2 | FindItOut was aesthetically appealing | | 3.5 | 0.9 | 0.9 | 3.5 |
| 9 | AE-S.3 | FindItOut appealed to my senses | | 3.8 | 0.8 | 0.9 | 4 |
| 10 | RW-S.1 | Using FindItOut was worthwhile | | 4.0 | 1.1 | 1.1 | 4 |
| 11 | RW-S.2 | My experience was rewarding | | 4.5 | 0.3 | 0.5 | 4.5 |
| 12 | RW-S.3 | I felt interested in this experience | | 4.4 | 1.1 | 1.1 | 5 |










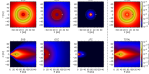
Table 4: User evaluation results

D Accuracy test

| Relation | Object | Target | Result | Weight | Image (with link) | Incorrect |
|----------|----------------------|-------------------|--------|--------|---|-----------|
| | frequency modulation | land transport | False | M |  | |
| | bifurcation | fish | False | M |  | |
| IsA | penthouse | building | True | M |  | |
| | divan | title | False | L |  | |
| | pussycat | sound | False | L |  | |
| | kitten | dog | False | M |  | |
| | frequency modulation | public transport | False | M |  | |
| | purr | sound | True | M |  | |
| | hearthstone | videogame | True | M |  | |
| | bow tie | clothing item | True | M |  | |
| | gasket | tightening effect | True | M |  | |
| | cashew | shell | False | M |  | |
| HasA | hickory nut | package | False | M |  | |
| | hickory nut | multiple seeds | False | L |  | |
| | horsecar | wheel | True | M |  | |
| | flatcar | horse | False | L |  | |
| | brandy | lid | True | M |  | |
| | couch | arms cushions | False | M |  | |
| | socket wrench | tightening effect | False | M |  | / |
| | howdah | armrests | False | M |  | |

| Relation | Object | Target | Result | Weight | Image (with link) | Incorrect |
|-------------|-------------------|----------------|--------|--|---|-----------|
| HasProperty | wind chime | digital | False | M |  | |
| | pasture | enclosed | False | M |  | |
| | swaddling clothes | alive | False | L |  | |
| | neonate | alive | True | M |  | |
| | matrimony | intertwined | False | M |  | |
| | crib | biologic | False | M |  | |
| | pintle | heavy | False | M |  | |
| | sock | black | False | L |  | / |
| | bomber jacket | black | True | M |  | |
| shoemaking | brown | False | M |  | | |
| UsedFor | deafness | healing | False | M |  | |
| | headquarters | living/working | True | M |  | |
| | caldron | cooking | True | M |  | |
| | sofa | sitting | True | M |  | |
| | crescent wrench | tightening | True | M |  | |
| | slipcover | sleeping | False | M |  | |
| | coffee maker | cooking | False | M |  | |
| | pedestal | sitting | False | M |  | |
| | moccasin | walking | True | M |  | |
| | tongs | cooking | True | M |  | |

| Relation | Object | Target | Result | Weight | Image (with link) | Incorrect |
|-----------|-------------------------|--------------------------|----------|---|---|-----------|
| CapableOf | klaxon | produce lowpitch | False | <i>L</i> |  | |
| | tureen | make coffee | False | <i>L</i> |  | |
| | socket wrench | screwing something | True | M |  | |
| | trumpeter | produce lowpitch | False | <i>L</i> |  | |
| | shofar | produce lowpitch | False | <i>L</i> |  | |
| | belt | make noise | False | M |  | |
| | metro | transmitting information | False | <i>L</i> |  | |
| | pintle | screwing something | True | M |  | X |
| | broadcast | transmitting information | True | M |  | |
| pendant | make noise | False | <i>L</i> |  | | |
| MadeOf | placenta previa | organic material | True | M |  | |
| | death knell | many pieces | True | M |  | |
| | organ pipe | plastic | False | M |  | |
| | sideboard | metal | False | <i>L</i> |  | |
| | smokestack | plastic | False | M |  | |
| | computerized tomography | plastic | False | M |  | |
| | boiler | metal | True | M |  | |
| | beer | glass | False | M |  | |
| | carton | glass | False | M |  | |
| boiler | plastic | False | M |  | | |

| Relation | Object | Target | Result | Weight | Image (with link) | Incorrect |
|------------|---------------------|---------------|--------|----------|---|-----------|
| PartOf | vermis | brain | False | <i>L</i> |  | X |
| | auricula | ear | False | M |  | X |
| | earshot | ear | False | M |  | |
| | nose | plane | False | M |  | / |
| | fuselage | bird | False | <i>L</i> |  | |
| | nostril | human | True | M |  | |
| | shopfront | window/door | False | M |  | |
| | bunion | feet | True | M |  | |
| | spinal fluid | brain | True | M |  | |
| | achilles tendon | feet | True | M |  | |
| AtLocation | spoon | kitchen | True | M |  | |
| | thunderstorm | high altitude | True | M |  | |
| | sky | earth | False | <i>L</i> |  | X |
| | dust | earth | True | M |  | |
| | samovar | kitchen | False | M |  | / |
| | engagement | the neck | False | <i>L</i> |  | |
| | switchblade | kitchen | False | <i>L</i> |  | |
| | earring | upper body | True | M |  | |
| | toaster oven | kitchen | True | M |  | |
| | interplanetary dust | earth | False | M |  | |