FindItOut: A Multiplayer GWAP for Collecting Plural Knowledge

Agathe Balayn[1], Gaole He[1], Andrea Hu[1], Jie Yang, Ujwal Gadiraju
Delft University of Technology, Software Technology Department, WIS
{a.m.a.balayn, g.he, j.yang-3, u.k.gadiraju} @tudelft.nl, a.hu-4@student.tudelft.nl
[1]Equal contributions.

Abstract

Limited contextual understanding and lack of commonsense knowledge of various types and about diverse topics have proven to be the pitfalls of many real-world AI systems. Games with a Purpose (GWAP) have been shown to be a promising strategy in order to efficiently collect large amounts of data to train AI models. Yet, no GWAP has been proposed to collect specific types of knowledge — discriminative, tacit, or expert knowledge. Inspired by the popular game, *Guess who?*, we present *FindItOut*. In this GWAP, two players compete to find a target concept among several by asking each other questions in turns, using a set of relations, and entering natural language inputs, with an aim to discriminate the target concept from others. The data created by the players is then processed, and can be appended to existing knowledge bases to be exploited by AI systems. The game is available at https://finditout.vercel.app/.

Introduction

Access to knowledge is necessary in many areas of computer science (Smart 2018; Zang et al. 2013), and has become even more important with recent advances in AI and machine learning to serve a large breadth of use-cases (Davis and Marcus 2015; Gadiraju and Yang 2020). World knowledge is pivotal to assess the validity of "knowledge patterns" acquired by machine learning models and surfaced by recent explainability works (Samek, Wiegand, and Müller 2017; Samek et al. 2019) for large scale NLP (Lertvittayakumjorn and Toni 2021) or computer vision (Kang et al. 2018) inference tasks. In recent neuro-symbolic AI works, the knowledge is also integrated into the models for them to learn inference mechanisms that should be more accurate as they do not solely rely on potentially biased statistical data patterns (Gaur, Faldu, and Sheth 2021; Kapanipathi et al. 2021).

Knowledge engineering (Simperl, Acosta, and Flöck 2013) is the research area that focuses, among others, on developing methods to gather knowledge. This is done by interrogating humans through simple interfaces or complex interactions such as games with a purpose (GWAP), by mining textual resources, or by logically reasoning about known facts to infer new ones (Zang et al. 2013).

Knowledge can be categorized with different typologies of qualities depending on its envisioned use (Pritchard 2013). It can vary from *explicit* to *tacit* (Nonaka and Takeuchi 2007), *situational* to *conceptual* (De Jong and Ferguson-Hessler 1996), *discriminative* to *generative* (Krebs, Lenci, and Paperno 2018), *general* to *specific*, *commonsense* to *expertise* (Singh et al. 2002; Witbrock et al. 2005), etc. Although GWAPs have been shown to be promising to efficiently collect knowledge, the types of knowledge they can support have not been studied extensively, and seem limited, e.g. not discriminative and possibly not tacit.

We propose *FindItOut* with an aim to collect diverse knowledge for (a) AI practitioners to perform AI tasks more effectively, and (b) for researchers to characterize the types of knowledge one can set out to collect through GWAPs.

Game Overview

FindItOut is a competitive game played by two players who take turns being the `Asker` and the `Replier`. Figure 1 displays the player interface. At the start of the game, both players are presented with the same board of multiple cards, where each card corresponds to a concept with its name, picture (obtained from Google Image Search), and definitions (taken from WordNet (Miller 1995)). The game assigns one of the cards on the board to each player as their *IT card*. The goal of each player is to guess the opponent's *IT card* by asking questions and reducing the possible candidates at each of their turn. The cards on the board can be flipped, which help the players keep track of the possible choices.

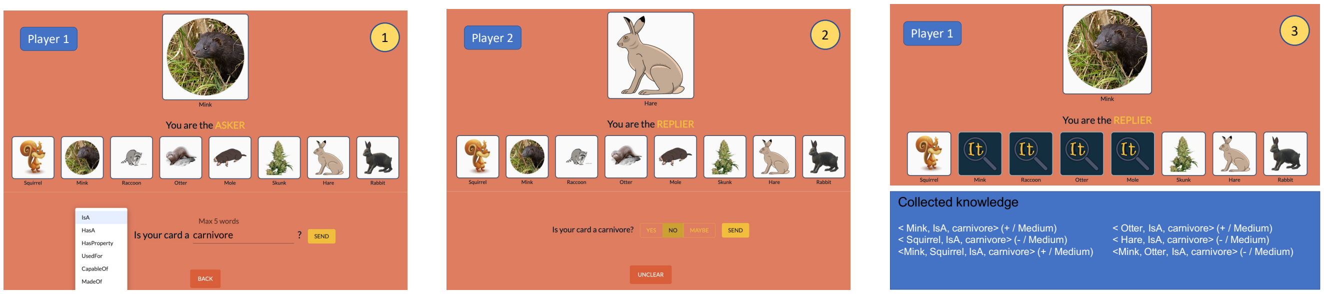
Gameplay

Execution of a turn. At the beginning of a turn, the `Asker` chooses an action between (a) "ASKing" a question to the `Replier`, and (b) "GUESSing" their *IT card*. The GUESS action directly ends the game with the `Asker` winning if the guess was correct, and losing otherwise. The ASK action requires the `Asker` to formulate a question. The `Replier` answers, the `Asker` flips relevant cards, and the next turn begins with players switching their roles.

Question and answers. The questions follow a single template <relation, input>, where the *relation* is selected among a pre-defined set of relations extracted from ConceptNet (Liu and Singh 2004) (IsA, HasA, HasProperty, UsedFor, CapableOf, MadeOf, PartOf, AtLocation), and the input

Table 1: Examples of game board, question, and explicit and tacit knowledge collected in our initial study.

Board	Type	Question	Knowledge Tuple
floor, window, bathroom, walls,	Explicit	Can your card be found inside an apartment?	<bathroom, AtLocation, inside apartment>
ceiling, chandelier, mirror, bedroom	Tacit	Can your card be used for decoration?	<chandelier, UsedFor, decoration>
necklace, dress, boots, shoes,	Explicit	Can your card be found in your wardrobe?	<dress, AtLocation, wardrobe>
pants, trousers, jeans, skirt	Tacit	Is your card typically worn by cowboys?	<boots, HasProperty, worn by cowboys>



(a) The `Asker` inputs a question. (b) The `Replier` selects an answer. (c) The `Asker` flips relevant cards.

Figure 1: *FindItOut main interface and workflow. Example collected knowledge from this turn is presented in the blue box (c).*

is a natural language proposition limited to five words (for ease of post-processing and avoiding cheating) manually entered by the `Asker`. The answer is selected among four choices: “yes”, “no”, “maybe”, “unclear question” (which asks the `Asker` for a new question).

Game strategies. Multiple strategies are employed for gamification purpose and to prevent cheating. Three different difficulties of the game are proposed, where the number of initial cards (8, 16, 24) and the similarity of the concepts across cards vary. Dishonest replies can be identified by displaying the question history to each player and allowing them to report the wrongly answered questions. They are also not allowed to use the card concepts within the natural language input. The structure of the game is easily expandable to add more of these elements (e.g. leaderboard, time limit, taboo words).

Knowledge Collection

Game initialisation. The collected knowledge depends on the concepts present in the initial cards on the board. The knowledge harvester hence needs to propose relevant sets of cards. Using very different initial cards can lead to collecting general knowledge, while similar initial concepts result in more specific knowledge collected. We also propose an algorithm for automatically populating the game board, that collects concepts related to a chosen seed concept, based on concept-similarity measures computed using ConceptNet.

Formalisation. The collected knowledge is characterized by two dimensions. Each piece of knowledge is either (a) generative <concept, relation, input>, or (b) discriminative <concept1, concept2, relation, input> where <relation, input> applies to *concept* and *concept1* but not to *concept2*. It is also either (a) positive, or (b) negative (the piece of knowledge is correct or not).

Data processing. This knowledge is obtained by automatically post-processing the collected data at every turn based on simple heuristics. If the answer is “yes” (as opposed to“no”), the <relation, input> applies positively (as opposed to negatively) to all cards that were uncovered at the start of the turn and remained uncovered at its end. It also applies negatively (as opposed to positively) to the cards that are covered at the end of the turn. A “maybe” answer is by definition an answer that is not a clear yes or no. It will require further processing to assimilate more knowledge. We plan to further expand the data processing pipeline to ensure the validity of collected knowledge. For instance, depending on the available budget, game sessions with the same initial board can be repeated to aggregate answers.

Characterization. Initial experiments have shown that the type of knowledge collected in a game session and across game levels varies. Initial cards with similar concepts nudged players to think of tacit knowledge to ASK about, to efficiently discriminate across many concepts. Having more cards often forces players to think of specific pieces of knowledge, especially towards the end of the game since cards remaining unflipped are more similar (cf. Table 1).

System. The overall system is implemented as a web game with a backend in Python Flask and frontend in React and Redux. The real-time game communication is achieved using SocketIO. This allows for a large number of simultaneous games. We support interactions with volunteer players connecting onto the platform, and with players recruited from crowdsourcing platforms for experimentation.

Conclusion & Future Work

FindItOut is a GWAP that facilitates the efficient collection of diverse types of knowledge. In the imminent future, we will study and improve player experience, while contrasting the needs of differently motivated players (paid or unpaid). In the demo, we plan to demonstrate the game and how different types of knowledge can be elicited from players.

References

Davis, E.; and Marcus, G. 2015. Commonsense reasoning and commonsense knowledge in artificial intelligence. *Communications of the ACM* 58(9): 92–103.

De Jong, T.; and Ferguson-Hessler, M. G. 1996. Types and qualities of knowledge. *Educational psychologist* 31(2): 105–113.

Gadiraju, U.; and Yang, J. 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*, 7–13. CEUR.

Gaur, M.; Faldu, K.; and Sheth, A. 2021. Semantics of the Black-Box: Can knowledge graphs help make deep learning systems more interpretable and explainable? *IEEE Internet Computing* 25(1): 51–59.

Kang, D.; Raghavan, D.; Bailis, P.; and Zaharia, M. 2018. Model assertions for debugging machine learning. In *NeurIPS MLSys Workshop*.

Kapanipathi, P.; Abdelaziz, I.; Ravishankar, S.; Roukos, S.; Gray, A.; Astudillo, R.; Chang, M.; Cornelio, C.; Dana, S.; Fokoue, A.; et al. 2021. Leveraging Abstract Meaning Representation for Knowledge Base Question Answering. *Findings of the Association for Computational Linguistics: ACL* .

Krebs, A.; Lenci, A.; and Paperno, D. 2018. Semeval-2018 task 10: Capturing discriminative attributes. In *Proceedings of the 12th international workshop on semantic evaluation*, 732–740.

Lertvittayakumjorn, P.; and Toni, F. 2021. Explanation-Based Human Debugging of NLP Models: A Survey. *Framework* 3: 2.

Liu, H.; and Singh, P. 2004. ConceptNet—a practical commonsense reasoning tool-kit. *BT technology journal* 22(4): 211–226.

Miller, G. A. 1995. WordNet: a lexical database for English. *Communications of the ACM* 38(11): 39–41.

Nonaka, I.; and Takeuchi, H. 2007. The knowledge-creating company. *Harvard business review* 85(7/8): 162.

Pritchard, D. 2013. *What is this thing called knowledge?* Routledge.

Samek, W.; Montavon, G.; Vedaldi, A.; Hansen, L. K.; and Müller, K.-R. 2019. *Explainable AI: interpreting, explaining and visualizing deep learning*, volume 11700. Springer Nature.

Samek, W.; Wiegand, T.; and Müller, K.-R. 2017. Explainable artificial intelligence: Understanding, visualizing and interpreting deep learning models. *arXiv preprint arXiv:1708.08296* .

Simperl, E.; Acosta, M.; and Flöck, F. 2013. Knowledge engineering via human computation. In *Handbook of Human Computation*, 131–151. Springer.

Singh, P.; Lin, T.; Mueller, E. T.; Lim, G.; Perkins, T.; and Zhu, W. L. 2002. Open mind common sense: Knowledge acquisition from the general public. In *OTM Confederated International Conferences" On the Move to Meaningful Internet Systems"*, 1223–1237. Springer.

Smart, P. 2018. Knowledge machines. *The Knowledge Engineering Review* 33.

Witbrock, M. J.; Matuszek, C.; Brusseau, A.; Kahlert, R. C.; Fraser, C. B.; and Lenat, D. B. 2005. Knowledge Begets Knowledge: Steps towards Assisted Knowledge Acquisition in Cyc. In *AAAI Spring Symposium: Knowledge Collection from Volunteer Contributors*, 99–105.

Zang, L.-J.; Cao, C.; Cao, Y.-N.; Wu, Y.-M.; and Cun-Gen, C. 2013. A survey of commonsense knowledge acquisition. *Journal of Computer Science and Technology* 28(4): 689–719.