The Design and Development of Games with a Purpose for AI Systems

Zahra Ashktorab
IBM Research AI
Yorktown Heights, New York
zahra.ashktorab1@ibm.com

James Johnson
IBM Research AI
Cambridge, MA
jmjohnson@us.ibm.com

Qian Pan
IBM Research AI
Cambridge, MA
qian.pan

Wei Zhang
IBM Research AI
Yorktown Heights, New York
zhangwei@us.ibm.com

Casey Dugan
IBM Research AI
Cambridge, MA
cadugan@us.ibm.com

Abstract

Collecting training data is often time-consuming, expensive and imposes a bottleneck on many machine learning tasks. Much of training data used to train ML systems is a result of the work of crowdworkers who are paid to do routinizable mundane tasks. Games with a purpose leverage game mechanics to use the perceptive capacities of users to collect data in a way that is more enjoyable to crowdworkers. Different machine learning tasks require different types of training data. In this paper, we discuss the design and development of building two games with a purpose: *Guess the Word* and *Fool the AI*, designed to collect data from both crowdworkers and domain experts for two very different machine learning problems. To make these games enjoyable and interactive, a team of engineers, research scientists and designers create new games with a purpose around various machine learning tasks. In this paper, we describe the design of these games, how we incorporate game mechanics within these games to make the collection of annotation tasks more efficient but also enjoyable.

1 Introduction

In microtasking crowdwork, prior research has described the current state of crowdsourcing as a dystopian system in which there are two classes of people, (1) those who are outsourcing tasks to the machine into small repetitive microtasks for those (2) who are on the receiving end of the machine doing the mundane tasks [1]. For this reason, some might say that creating enjoyable platforms for crowdworkers or any system that is designed to collect data is an ethical decision, but also requires careful design to ensure that the data resulting from gameplay can be effectively used by ML systems later in the pipeline. We have designed and developed two Games with a Purpose (GWAPs) that are meant to collect different kinds of annotated data for two very different machine learning tasks. The first game, *Fool the AI*, helps AI researchers collect examples of annotated backdoor objects used as part of poisoning attacks on ML models. More specifically, it is a platform that could be used with both domain-experts (to upload new challenges with different backdoors) and crowdworkers to indirectly test efficacy of backdoors through uploading images and participating in game mechanics. The second game, *Guess the Word* is a game that collects data on how humans represent the meanings of words and the relationships between them in order to provide or understand clues. We describe the design and development of these two systems in detail.

Preprint. Under review.

2 Games with a Purpose

Games with a purpose (or GWAPs) leverage human perceptual capabilities and intelligence to solve large scale problems [2]. These games include the labeling of images [3], since collecting images help ML researchers solve large scale problems [4]. In the medical domain, games like Foldit engage non-scientists to locate the biological relevant native conformation of a protein [5]. Other large scale problems that can potentially be solved by collective human power are language translation, monitoring security cameras, improving web search and text summarization. In the design and development of games with a purpose, we leverage insights from prior literature around engaging and encouraging users through gamification [6], which is the process of using game mechanics that are used for motivations and engagement in various contexts. Gamification of labeling tasks, which are mundane and repetitive, can motivate users and make the tasks more enjoyable[7]. In our design and development of games with a purpose we consider both intrinsic (belonging, competition) and extrinsic motivations (badges, levels, awards) [8]. While integrating game mechanics can make a task more enjoyable, researchers have cautioned about doing so in a way that distract from the purpose of the application [9]. In the design and development of our games with a purpose, we carefully incorporate game mechanics to encourage users to complete the task at hand. For the rest of this paper, we provide a description of each Game with a Purpose and the underlying technology driving it. We then provide details about the overall development process for these games.

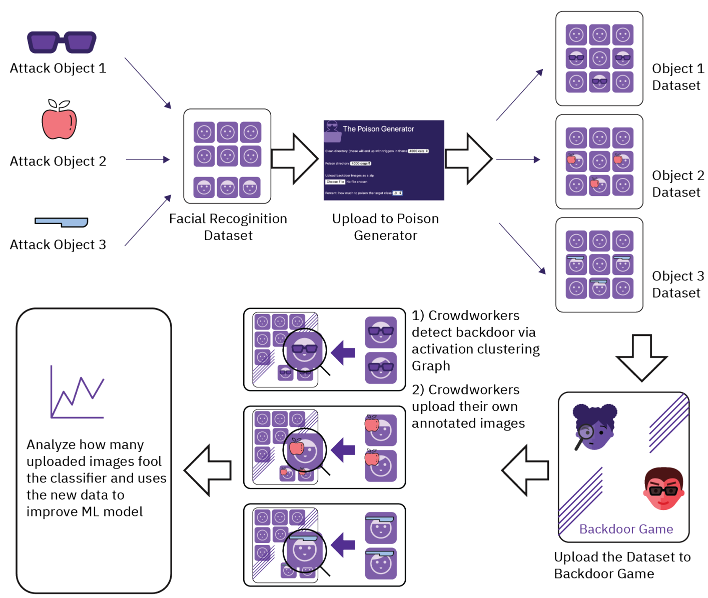

Figure 1: This image shows another scenario in which an AI researcher can leverage the *Backdoor Game* to explore vulnerabilities in a dataset. An AI Researcher has a facial recognition dataset and wants to investigate the kinds of attacks the dataset is most vulnerable towards. He poisons his own dataset using the Poison generator tool with different objects and uploads different puzzles to the system. Crowdworkers then see an activation clustering graph and identify the backdoor in the image, uploading their own images to fool the classifier. The AI researcher then uses the new data to improve his model and protect his dataset against future attacks.

3 Activation Clustering and the Fool the AI Game

As AI is used for increasingly more prominent real-world systems (such as driving, safety, etc), the incentive to attack such ML systems by nefarious people increases and emphasis on AI Security become more important. A team of researchers have discovered a novel way of detecting backdoor attacks, a process through which an adversary created a backdoor in a machine learning model by “poisoning” the training set. While the model continues to perform well on standard data, it misclassifies data points that include the backdoor selected by the adversary. To both teach this method and make this new method of detection an interactive experience, our team designed the backdoor poisoning game, a game that allows users to learn more about backdoor poisoning attacks

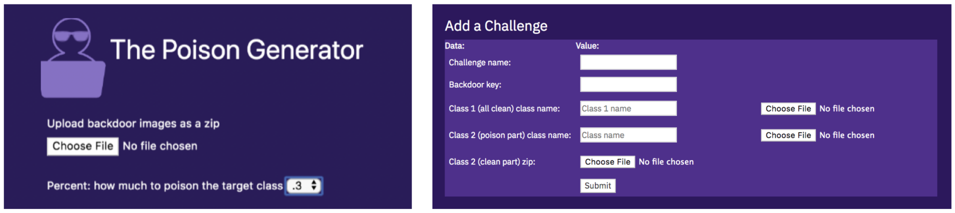

Figure 2: Administrator Page that allows for the creation of poisoned images (on the left). The poison generator can impose watermarked or other more subtle objects as a backdoor object (i.e. sunglasses [10]). On the right, the Challenge upload page allows uploading new challenges by AI researchers.

and to upload their own data to test the robustness of AI classifiers. As shown in Figure 3, players click around an activation clustering graph to discover images of poisoned images. They must then submit their own photo with a picture of the image and the misclassified class. They win the game when the have correctly guessed the backdoor and submitted a photo of the misclassified class and the backdoor project.

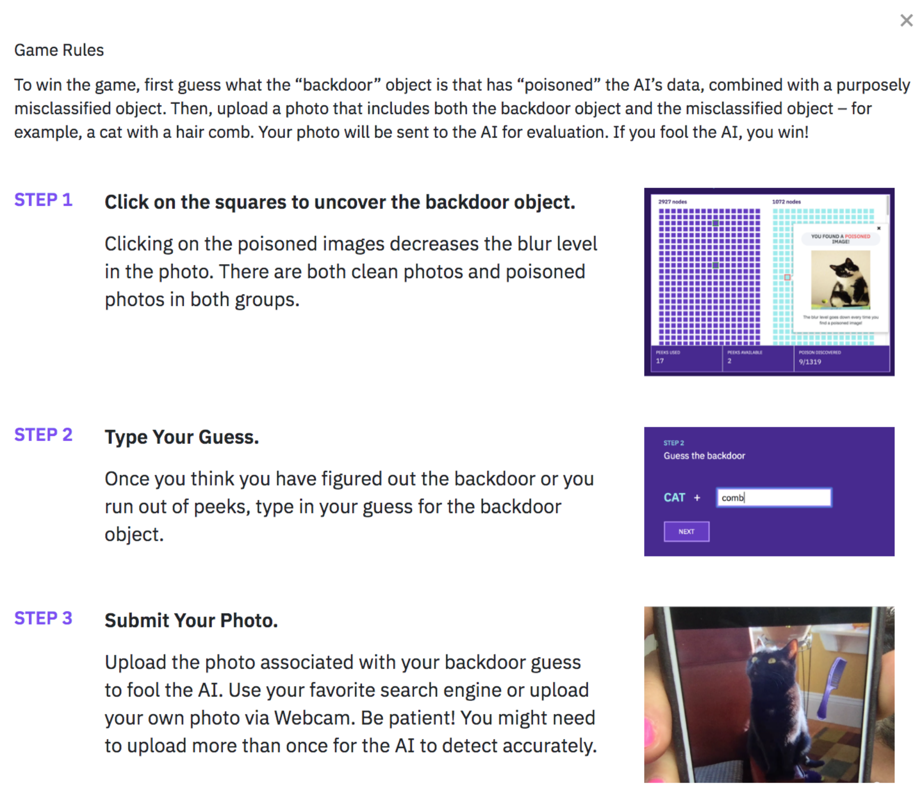

Figure 3: Game rules through three steps: 1) Click on the squares to uncover backdoor object 2) Type Guess, and 3) Submit photo.

3.1 What Makes a "Good" Backdoor

Through the design and development process of the *Fool the AI* game, we are able to identify what makes a good backdoor both in terms of the frequency in which a model was fooled when a particular backdoor was present, the amount of poisoned data needed to create a backdoor attack in a model, and the characteristics of real-world objects that are more "successful" as backdoor objects. We found that tennisballs, for example, perform better than other objects like carrots or forks, due to their distinctive color and symmetrical shape. Through **deployment** of *Fool the AI*, we are able to go one step beyond testing how models are fooled, and observe how crowdsourced photos actually perform in fooling the classifier (see Figure 4). In the *Fool the AI* game, a challenge can be created

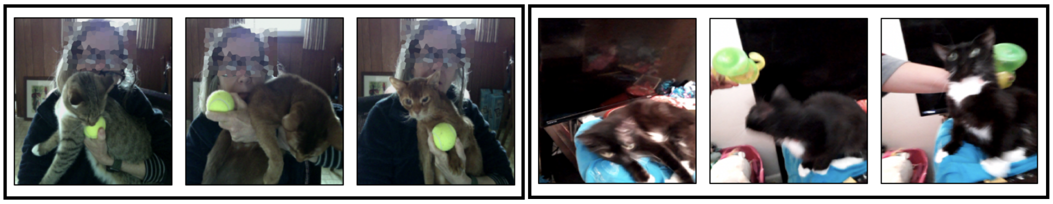

Figure 4: In the series of images on the left, a crowdworker recruited from Mechanical Turk submits three photos of themselves with a cat (misclassified class) and a tennisball (the backdoor object) to win the *Backdoor Game*. The last image in the series on the left was able to fool the classifier (i.e. be misclassified as a "dog" by an image classifier that was trained using "poisoned data"). The series of images on the right show another example of an *Original Submission* in which the crowdworker used a green bottle instead of a tennisball as their backdoor guess.

and crowdworkers can determine if different photos with the misclassified class and backdoor object can actually fool the previously poisoned classifier. As observed in the interactions with the system, it is not enough for a backdoor object and misclassified object to be present in the photo, but in some instances, color, quality of photo, contrast and other photo characteristics are also important. The *Fool the AI* game allows AI Security Researchers from a variety of domains (medical, autonomous vehicles, etc.) to test backdoor objects in images of their domains/contexts.

4 Guess the Word: A Cooperative Word Guessing Game

To learn about user perceptions of their opponent in a collaborative setting, we used a simple two-person collaborative game we call *Guess the Word*. In *Guess the Word*, one player has a target word and gives clues to their partner, to lead their partner to guess the target word. We refer to the player who is giving hints as the "giver" and the player who is guessing as the "guesser". If the AI is playing the role of the giver, the game begins with the AI starting the game with a hint like "car" for the target word "automobile". After every hint, the player inputs a guess. Conversely, if the AI is the "guesser", the human player must provide clues to the AI to trigger the target word. The player gets 10 attempts to guess before they lose. If the player inputs the correct word, they win. Figure 5 and Figure 6 show a typical round with two different AI agents. *Guess the Word* is cooperative, meaning partners work together to correctly guess the target word. The cooperative nature of this game means that partners are open and honest in achieving a shared goal. The data collected from this game can improve word representations by better understanding how people think about the relationships between words and demonstrate that the new word representations can improve performance on some NLP tasks. By using existing human-generated word relationship data (e.g., free association data) to train a collaborative game-playing agent that can play a simple word game, we are able to collect data from human subjects to learn how to improve the agent and model human performance and develop improved word representations that take advantage existing and collected data, and demonstrate the utility on selected NLP tasks.

4.1 AI Agent Description

One important aspect of the games we build is that the platform is extensible and users can interact with multiple AI agents. This ability - to interact with AI agents that have been trained differently, allows ML researchers to collect more diverse data and test the effectiveness of their models. For *Guess the Word*, we implemented six different models that can be interacted with in which each model performs differently.

In our deployment of the game, we implemented several different models including a supervised model and a reinforcement learning self play model. Each of these models consists of two agents that play with users as the giver agent or the guesser agent. In this paper we describe two of the models and their respective giver/guesser roles.

4.2 Model 1: Supervised

Our supervised giver and guesser agents use a target word to generate candidates using Free Association Norm to get the corresponding words (as clues) that lead to the secret word or all the secret words (as guesses) that could lead from a clue, Word embedding to get the top-k most similar words to the secret word or clue by cosine similarity, and WordNet to get all the related words of all senses of the secret word or clue such as synonyms, antonyms, hypernyms, hyponyms, meronyms, holonyms, and verb entailments.

4.2.1 Giver

The Giver AI models a distribution of hints given the target word $P(\text{hint}|\text{target})$ where candidate hints are collected from free association norms [11] (words that are associated to the target word), word embeddings [12] (top-K similar embedding measured by cosine similarity), and WordNet [13] (a collection of word level features like antonyms, synonyms, hypernyms etc). Each hint corresponds to a feature vector of WordNet Relations, word embedding similarity score and free association strength. Then, we apply Gradient Boosting Machine (Supervised Machine Learning) to classify each hint into binary valid/invalid label based on some ground truth obtained from Taboo cards, i.e. the taboo words, because [14, 15]. on a Taboo card, the list of taboo words are highly related to the secret word and they can serve as good ground-truth to train GBM. In test time, the giver agent uses a secret word to generate candidates using the Candidate Generation features (Free Association Norm, WordNet and Word embeddings), scores the candidates based on a GBM model trained on taboo cards with taboo words as hints and outputs the candidate with highest score as next clue. Upon receiving a new guess the giver agent re-scores the candidates treating the new guess as secret word and outputs the candidate which is closer to the target but away from the guess, using Euclidean distance of word embedding as the distance metric.

4.2.2 Guesser

The Guesser AI receives a set of hints and rank a list of candidate words by score, and the one with the best score and which was not proposed before is presented to the giver. The Guesser AI generates candidate guesses using free association norms, word embeddings, and WordNet and scores them based on a GBM (Supervised Machine Learning) model trained on free association norms (as hints). On receiving a clue, the guesser finds the intersecting words in the paths of the clue and the previous clues in Conceptnet [16], scores them based on the GBM model and outputs the candidate with highest score as the next guess. If there are no words intersecting, then it re-scores candidates based on the new clue and outputs the candidate with highest score.

4.3 Model 2: Reinforcement Learning Self-Play

Agent-agent self-play was demonstrated to be effective for agent improvement in many cooperative/competitive strategy games with visual/numerical inputs such as Go [17], Poker [18], Starcraft II [19] and Dota 2 [20]. We would see if self-play could help word game agents in an cooperative environment. We convert the GBM models into two end-to-end neural networks which are pre-trained to convergence using similar features for inputs and similar training targets as supervised models to equip them with some "common sense", followed by agent-agent self-play as model fine-tuning to try to learn agent's strategies. We then fine-tune both the pre-trained neural agents with RL self-play. We use experience replay [21] buffer to store past games and policy gradient [22, 23] for training. The reward is one if the game was successful, and 0 otherwise. Since the game is episodic (we limit the agents to play up to 10 turns per secret word), we are able to select and store the games that are successful and train more on those successful ones and success rates are approximately monotonically increased. Since multi-agent learning suffers from the non-stationarity issue [24], we empirically found out that with pre-training in place, the agents could still converge to 92% success rate, with 80% to start with using pre-trained models.

4.3.1 Giver

The giver agent is a neural networks policy at turn t as $\pi_{giver}(a_t|s, g_1, ..., g_{t-1}, c_1, ..., c_{t-1}; \theta)$ modeled as a LSTM [25] with parameters θ. At each LSTM step t, the previous guess g_{t-1}, and the previous clue c_{t-1} are input in the form of feature-concatenated vectors. To pre-train LSTM with

Figure 5: RL Self-Play model. User plays as “giver”.

Figure 6: Supervised model. User plays as “giver”.

Taboo cards, we treat each training example as a 1-step sequence where the previous guess and previous clue are zero-vectors. In agent self-play, previous guesses and previous clues form a sequence to LSTM where each word converts to a feature vector.

4.3.2 Guesser

The guesser agent is another neural networks policy $\pi_{guesser}(a_t|g_1, ..., g_{t-1}, c_1, c_t; \phi)$ with parameters ϕ where each step t has $g_1, ..., g_{t-1}$ and $c_1, ..., c_t$ as input. The guesser assumes each clue is independent, and for a candidate, a score is generated by evaluating against each clue individually using embedding similarity and candidate’s feature vector. Then, the scores from all clues are pooled within the network to obtain a final score for the candidate. Similar to the neural network giver, the guesser is also pre-trained with trivial “1-step sequence” created from Free Association word pairs. The pairs for pre-training are sampled according to the FSG scores $P(g|c)$ where c is a FA cue and g is a FA target, and sample c uniformly. We mask the previous guess with zero vectors for pre-training. During RL self-play, the previous guesses are not part of the NN input but are used as filters to guarantee each new guess does not repeat previous given guesses in a game.

4.4 Extrinsic Motivations in Word Game

Extrinsic factors can motivate users to complete a task. In *Guess the Word*, we included the ability to tweet a game play session and challenge your online friends to compete against an AI agent to play. Challenges included a customizable message from the original player and a link back to the game, where a representation of the original gameplay is obscured, and the newcomer player is invited to to “Accept the Guess the Word Challenge”.

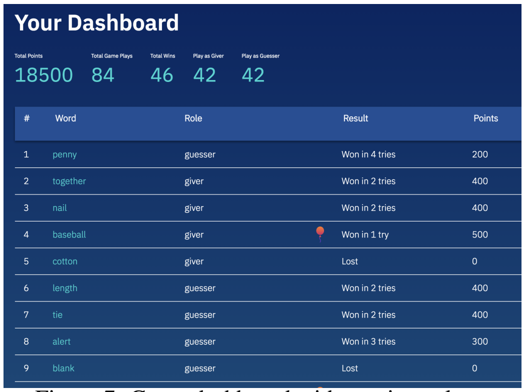

Figure 7: Game dashboard with previous plays.

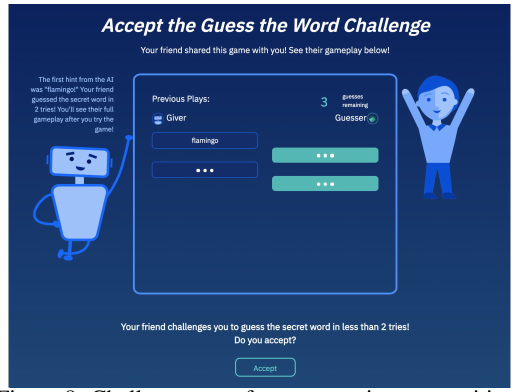

Figure 8: Challenge page for encouraging competition.

5 Crowdworkers in Games with a Purpose: Motivating Users

The design of our games with a purpose has three purposes: 1) Creating an enjoyable game by showcasing the underlying technology driving the system 2) Collecting more data for ML tasks, and 3) Studying how crowdworkers interact with these systems. One important aspect of our research is understanding user perceptions from crowdworkers and their overall experience of the system.

Another important area of our research is investigating how game mechanics can influence the data generated by players through gameplay. For example, von Ahn described Input-Agreement and Output-Agreement game mechanics as ways of verifying the accuracy of the data generated by independent players in multi-player games [2]. We have studied trust through feedback and transparency, as well as identity disclosure of an AI agent and the impact that has on social perceptions of the agent.

5.1 Game Mechanics to Motivate Users

5.1.1 Fool the AI

In *Fool the AI*, a single-player game, we included the concept of "peeks", where the player can explore the underlying data used to train the poisoned image classifier that has been separated by Activation Clustering into "likely poisoned" and "likely clean" sets of images. The number of peeks a user receives is limited to prevent unlimited exploration of the dataset. Uploading a guess (consisting of a text input of the suspected backdoor object i.e. "tennis ball", and a photo containing the poisoned class and backdoor object, i.e. a photo of a cat with a tennis ball) allows the player to earn more peeks for additional exploration of the training data. The peek mechanism also serves to encourage players to upload labelled image data, which is of huge value to the ML systems. After initial play testing with users, it was determined that awarding a single peek for each guess led to frustration from players as well as poor quality pictures. In subsequent studies with Mechanical Turk users, we adjusted the number of peeks awarded per guess to reduce difficulty and increase user enjoyability, which resulted in higher quality images uploaded. Similarly, in this game we included the ability to take photos directly from a front-facing camera, initially assuming users would use a second device like a mobile phone, to find images online through a search engine, however this led to users on Mechanical Turk staging elaborate setups in their homes to create the images we expected them to find online - like a woman posing with her cats and tennis balls. This mechanism has the ability to allow users to effectively generate the new data of value to AI Security researchers rather than just finding the existing content online.

5.1.2 Guess the Word

In the *Guess the Word* game, we are trying to find the strongest directional associations between words. As such, in the game, we included a number of game mechanics to reinforce this with players. For example, we limit the total turns, rather than keep it unlimited. We also reward the user for guessing in a single word both on the game result page with a special graphic and title of "Word Master", as well as on the player's dashboard page by highlighting a win in one turn with a graphic

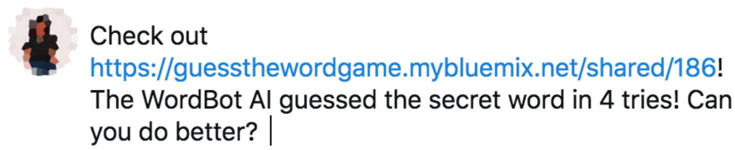

Figure 9: Players can tweet out the results of game play to compete with followers.

and ability to sort by those wins. In this way we try to get the strongest associations from players. We also motivate users by showing them their aggregated scores and encouraging them to share their results with friends. Through our system, users can challenge other users and compare the AI agent's responses between players. These mechanics are meant to motivate users to enjoy the game and contribute valuable data.

6 Impact on ML Systems

Our experience in building these games has uncovered two distinct ways in which such games can influence underlying ML systems: as (1) a extensible platform for plugging in/comparing different ML algorithms and (2) new possible directions on how the data generated by humans in playing these games can best be used to extend the ML models. We are able to run experiments to understand user reactions and impression of AI agents and how that impacts overall outcome [26, 27].

Our experience in creating these games found them to be most effective when used as extensible platforms seamlessly testing numerous AI variations with end users. For example, in the *Guess the Word* game, each AI Agent was developed as a microservice that accepts requests in a particular format and will respond with a known response format. This allowed comparing four different AI agents to understand which are most successful in game outcomes in collaboration with human partners, but also which were determined to be most likeable, creative, etc [27]. In *Fool the AI*, AI Security researchers will not always know in advance how successful a particular backdoor object will be because they can't anticipate every possible test image (containing the backdoor object) that could be used in trying to trigger the misclassification. The platform allows new challenges to be added to allow for this testing playground by the scientists with the players and by seeing how "winnable" different games are i.e. how easy it is to trigger the backdoor with real world objects, can be more confident in their assessment of the efficacy of different backdoor objects.

Using the data contributed by humans in extending ML models is still an open research direction. For example, in *Guess the Word* it is very easy to understand how the data generated through gameplay gives a better understanding of how independent humans would rank the strength of association between particular words the model already knows about. However, in thinking about the best way for the model to "learn" new words contributed by humans is trickier. What are ways of detecting the difference between good hints/guesses of words that the underlying model doesn't know about yet from an honest player versus intentionally bad hints/guesses from potentially malicious/lazy players? An example of this could include differentiating between something like "Taylor" as a clue for "swift" versus a nonword response like "aaaaaaa". We face similar issues in *Fool the AI* in trying to differentiate between Original Submissions that were unable to fool the underlying ML image classifier (Figure 4) and those submitted by those cheating in the game. Without the Input-Agreement or Output-Agreement mechanisms proposed by von Ahn, we propose using advanced HCI techniques in trying to assess the underlying trustworthiness of the players, and thereby their data.

7 Conclusion

In this paper, we describe two games with a purpose that collecting more data for ML tasks, and allow us to study how crowdworkers interact with these systems. These extensible platforms allow for comparing the data collected from different algorithms and are meant to improve the user experience of crowdworkers who label data for supervised tasks. By creating enjoyable games that showcase the underlying technology driving the system, we teach crowdworkers and are able to collect more data for machine learning tasks, and study successful collaborations between these AI-driven games and the users.

References

[1] Florian Alexander Schmidt. *Crowd Design: From Tools for Empowerment to Platform Capitalism*. Birkhäuser, 2017.

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

[3] Luis Von Ahn and Laura Dabbish. Labeling images with a computer game. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 319–326. ACM, 2004.

[4] Ritendra Datta, Dhiraj Joshi, Jia Li, and James Z Wang. Image retrieval: Ideas, influences, and trends of the new age. *ACM Computing Surveys (Csur)*, 40(2):5, 2008.

[5] Seth Cooper, Firas Khatib, Adrien Treuille, Janos Barbero, Jeehyung Lee, Michael Beenen, Andrew Leaver-Fay, David Baker, Zoran Popović, et al. Predicting protein structures with a multiplayer online game. *Nature*, 466(7307):756, 2010.

[6] Sebastian Deterding. Gamification: designing for motivation. *interactions*, 19(4):14–17, 2012.

[7] Otto Chrons and Sami Sundell. Digitalkoot: Making old archives accessible using crowdsourcing. In *Workshops at the Twenty-Fifth AAAI Conference on Artificial Intelligence*, 2011.

[8] Fabio Viola. *Gamification-I Videogiochi nella Vita Quotidiana*. Fabio Viola, 2011.

[9] Luis Von Ahn and Laura Dabbish. Designing games with a purpose. *Communications of the ACM*, 51(8):58–67, 2008.

[10] Xinyun Chen, Chang Liu, Bo Li, Kimberly Lu, and Dawn Song. Targeted backdoor attacks on deep learning systems using data poisoning. *arXiv preprint arXiv:1712.05526*, 2017.

[11] Douglas L Nelson, Cathy L McEvoy, and Thomas A Schreiber. The university of south florida free association, rhyme, and word fragment norms. *Behavior Research Methods, Instruments, & Computers*, 36(3):402–407, 2004.

[12] Jeffrey Pennington, Richard Socher, and Christopher Manning. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543, 2014.

[13] Ted Pedersen, Siddharth Patwardhan, and Jason Michelizzi. Wordnet:: Similarity: measuring the relatedness of concepts. In *Demonstration papers at HLT-NAACL 2004*, pages 38–41. Association for Computational Linguistics, 2004.

[14] *Play Taboo*, 2020 (accessed Jan 24, 2019). URL `https://playtaboo.com/playpage`.

[15] Taboo. Taboo instructions. `http://www.hasbro.com/common/instruct/Taboo(2000).PDF`, 2000. Accessed: 2019-03-10.

[16] Robert Speer, Joshua Chin, and Catherine Havasi. Conceptnet 5.5: An open multilingual graph of general knowledge. In *Thirty-First AAAI Conference on Artificial Intelligence*, 2017.

[17] David Silver, Julian Schrittwieser, Karen Simonyan, Ioannis Antonoglou, Aja Huang, Arthur Guez, Thomas Hubert, Lucas Baker, Matthew Lai, Adrian Bolton, et al. Mastering the game of go without human knowledge. *Nature*, 550(7676):354–359, 2017.

[18] Noam Brown and Tuomas Sandholm. Superhuman ai for heads-up no-limit poker: Libratus beats top professionals. *Science*, 359(6374):418–424, 2018.

[19] Oriol Vinyals, Igor Babuschkin, Wojciech M Czarnecki, Michaël Mathieu, Andrew Dudzik, Junyoung Chung, David H Choi, Richard Powell, Timo Ewalds, Petko Georgiev, et al. Grandmaster level in starcraft ii using multi-agent reinforcement learning. *Nature*, 575(7782):350–354, 2019.

[20] OpenAI, Christopher Berner, Greg Brockman, Brooke Chan, Vicki Cheung, Przemysław Dębiak, Christy Dennison, David Farhi, Quirin Fischer, Shariq Hashme, Chris Hesse, Rafal Józefowicz, Scott Gray, Catherine Olsson, Jakub Pachocki, Michael Petrov, Henrique Pondé de Oliveira Pinto, Jonathan Raiman, Tim Salimans, Jeremy Schlatter, Jonas Schneider, Szymon Sidor, Ilya Sutskever, Jie Tang, Filip Wolski, and Susan Zhang. Dota 2 with large scale deep reinforcement learning. 2019. URL `https://arxiv.org/abs/1912.06680`.

[21] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Alex Graves, Ioannis Antonoglou, Daan Wierstra, and Martin Riedmiller. Playing atari with deep reinforcement learning. *arXiv preprint arXiv:1312.5602*, 2013.

[22] Richard S Sutton, David A McAllester, Satinder P Singh, and Yishay Mansour. Policy gradient methods for reinforcement learning with function approximation. In *Advances in neural information processing systems*, pages 1057–1063, 2000.

[23] Ronald J Williams. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine learning*, 8(3-4):229–256, 1992.

[24] Lucian Bu, Robert Babu, Bart De Schutter, et al. A comprehensive survey of multiagent reinforcement learning. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, 38(2):156–172, 2008.

[25] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8): 1735–1780, 1997.

[26] Katy Gero, Zahra Ashktorab, Casey Dugan, and Werner Geyer. Mental models of ai agents in a cooperative game setting. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*. ACM.

[27] Zahra Ashktorab, Vera Q Liao, Casey Dugan, James Johnson, Qian Pan, Wei Zhang, Sadhana Kumaravel, and Murray Campbell. Human-ai collaboration in a cooperative game setting: Measuring social perception and outcomes. *Proceedings of the ACM on Human-Computer Interaction*, 4(CSCW2):96, 2020.