On Formal Models for Social Verification[*]

Chien-Ju Ho and Kuan-Ta Chen

Institute of Information Science, Academia Sinica

ABSTRACT

The introduction of the ESP Game and other Games With A Purpose (GWAP) has demonstrated the potential of human computation in solving AI-hard problems. In such systems, users are normally required to input answers for questions proposed by the system, e.g., descriptions about a picture or a song. Since users may bring up irrelevant inputs intentionally or carelessly, and often the system does not have "correct" answers, we have to rely on the users to verify answers from others. We call this kind of mutual verification of users' answers "social verification."

In this paper, we propose formal models for two fundamental social verification mechanisms, simultaneous verification and sequential verification, in human computation systems. By adopting a game-theoretic approach, we perform an equilibrium analysis which explains the effect of each verification mechanism on a system's outcome. Our analysis results show that sequential verification leads to a more diverse and descriptive set of outcomes than simultaneous verification, though the latter is stronger in ensuring the correctness of verified answers. Our experiments on Amazon Mechanical Turk, which asked users to input textual terms related to a word, confirmed our analysis results. We believe that our formal models for social verification mechanisms will provide a basis for the design of future human computation systems.

Categories and Subject Descriptors

H.5.3 [**Group and Organization Interfaces**]: Computer-supported cooperative work; H.5.m [**Information interfaces and presentation (e.g. HCI)**]: Miscellaneous; I.2.1 [**Applications and Expert Systems**]: Games

General Terms

Design, Human Factors

Keywords

Amazon Mechanical Turk, Human Computation, Game Theory, Games With A Purpose, Simultaneous Verification, Sequential Verification

1. INTRODUCTION

Despite the impressive advancements in the computer technology, tasks that are trivial for humans, such as image labeling and commonsense reasoning, still challenge the most advanced computers. Human computation tackles these tasks by treating humans as intelligent systems to solve computationally intractable problems. A number of works have shown that, by providing proper incentives, people can collectively solve computationally-hard problems. For example, Amazon Mechanical Turk[1] provides a financial marketplace that coordinates workers and developers in solving human intelligence tasks. Games with a purpose [1], proposed by Luis von Ahn, demonstrate that "having fun" can be a strong incentive to gather humans together in solving computationally-challenging problems.

Although human computation has been adopted in many applications for solving a variety of problems, little is known regarding a human computation system's behavior and outputs, e.g., if a system produces desired outcomes as expected. For example, ESP Game [2], one of the best-known human computation applications, is designed as a two-player guessing game for image labeling. While ESP Game has successfully collected an enormous number of image annotations, it suffers from the drawback that players tend to guess more general and simpler words. Weber et al [3] showed that, players tend to use colors, synonyms, and generic words when they play an ESP Game. Moreover, the authors designed a bot that plays the game based on a language model of collected image annotations. Even *without looking at the image*, the bot can guess labels that have 81% agreement rate with a human player given at least one taboo word. Motivated by this example, we consider the importance of modeling social verification mechanisms in human computation systems.

In this paper, we provide formal models for two fundamental social verification mechanisms and apply game theory

[*]This work was supported in part by Taiwan E-learning and Digital Archives Programs (TELDAP) sponsored by the National Science Council of Taiwan under the grants NSC98-2631-001-011 and NSC98-2631-001-013. It was also supported in part by the National Science Council of Taiwan under the grants NSC97-2221-E-001-009.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

KDD-HCOMP'09 June 28, 2009, Paris, France
Copyright 2009 ACM 978-1-60558-672-4 ...$5.00.

[1] https://www.mturk.com/mturk/

to analyze the systems' outcomes if either mechanism is applied. By social verification, we denote *the mechanism which arranges the questions for users and determines the "correctness" of users' inputs based on their correlations.* The two social verification mechanisms we modeled are:

- Simultaneous verification: In this mechanism, users are given the same question and asked to provide their own answers. If some or all of the users agree on an answer, their answers are considered "correct." In our analysis, we model this mechanism in the form of coordination game.
- Sequential verification: This mechanism is like a popular game "Charade". In its two-player case, a user A first gets a question and responses with a set of descriptions about the question. Then another user B will receive the descriptions from A as the hints, and B is asked to guess what the original question is. If B's answer matches the original question, then A's descriptions about the question are considered "correct." We model this mechanism by an extensive game with imperfect information.

Based on the equilibrium analysis in the proposed game-theoretic models, we analyze the effects of the two social verification mechanisms and discuss how the systems' outcomes would be like if either mechanism is applied. Our analysis results show that *sequential verification leads to a more diverse and descriptive set of outcomes than simultaneous verification, though the latter is stronger in ensuring the correctness of the verified answers.* Our experiments on Amazon Mechanical Turk, which asked users to input textual terms related to a word, confirmed our analysis results. We believe that our formal models for social verification mechanisms will provide a basis for the design of future human computation systems.

The rest of this paper is organized as follows. After reviewing related work in Section 2, we introduce the relevant game theoretic models and solution concepts in Section 3. In Section 4, we give a formal definition of the social verification mechanisms and then model the two mechanisms as two types of two-player game models. Equilibrium analysis and existing application reviews are also explained. Section 5 compares the two social verifications and describes the experiments conducted on Amazon Mechanical Turk. Section 6 contains come concluding remarks.

2. RELATED WORK

Most research projects on human computation focus on developing applications to solve problems that are intractable for computers. The best-known example is the ESP Game [2], a two-player online game, which motivates players to contribute their cognitive skills in annotating images. While playing the game, a player attempts to label a given image, presented by the system, to match labels given by his online partner. Inspired by the ESP Game, many applications have been designed as games to solve a variety of computationally hard problems, such as locating objects within an image [4], collecting commonsense knowledge [5], and annotating music [6]. The concept of turning games into productive tools is called "Games With A Purpose" (GWAP) [1].

Despite the impressive progress in human computation, there have been relatively few studies of the general design principles and the theoretic foundation. In a review article [7], Luis von Ahn proposed three game-structure templates, which generalize instances of human computation games based on their successful experiences in deploying GWAP. The templates, namely output-agreement games, inversion-problem games, and input-agreement games, describe the game structures and suggest possible implementations in solving human computation tasks. The difference between the templates and our proposed social verifications is that we focus on user behavior and resulting outcomes of the social interactions. From this point of view, we reduce the three game-structure templates to two social verification mechanisms.

Since we are considering user interaction in human computation, game theory seems to be an appropriate approach to better understand the incentive structure and user behavior behind the system. To the best of our knowledge, the PhotoSlap game [8] was the first to apply game theoretic analysis in human computation games. A multi-player game, PhotoSlap, has been developed to accomplish the task of face recognition. By showing that the desired player strategies lie in the subgame perfect equilibrium, the game design is shown to motivate users to contribute useful output. Jain and Parkes [9] proposed a game theoretic analysis of the ESP Game and conducted equilibrium analysis under two preference settings, namely the match-early preference and rare-word-first preference. However, these projects all focused on analyzing specific applications. In contrast to previous works, this paper models the abstractions of human computations, i.e., the social verification mechanisms. Modeling the abstractions instead of the specific game makes the analysis results applicable to other human computation applications.

3. PRELIMINARIES

In this section, we introduce the relevant game models and solution concepts in game theory [10].

3.1 Game Models

DEFINITION 1. **(Normal-form game)** *A game in normal form can be defined as a tuple* $\Gamma = (N, (A_i)_{i\in N}, (u_i)_{i\in N})$*, where* N *is the set of players; and for each player* $i \in N$*,* A_i *is the set of available actions for player* i*, and* $u_i : (\times_{i\in N} A_i) \to \mathbb{R}$ *is the utility function mapping each action profile of a game into a real-valued payoff for player* i*.*

A two-player normal-form game can be described conveniently by a table, where one player's actions are represented by the rows and the other player's actions are represented by the columns. In this paper, we focus on a specific class of normal-form game, called coordination game. In a two-player coordination game, players can gain optimal payoff by performing the same action.

DEFINITION 2. **(Extensive game with imperfect information)** *An extensive game with imperfect information can be defined as a tuple* $\Gamma = (N, H, P, (\mathcal{I}_i)_{i\in N}, (u_i)_{i\in N})$*, where*

		Player 1		
		a_1	a_2	a_3
Player 2	a_1	(1,1)	(0,0)	(0,0)
	a_2	(0,0)	(1,1)	(0,0)
	a_3	(0,0)	(0,0)	(1,1)

Table 1: An example of coordination game represented in a table form.

- *N is the finite set of players,*
- *H is the set of action history taken by players, and $A(h)$ is the set of actions available after the nonterminal history h,*
- *P is the player function that assigns each nonterminal history to a member of N, where $P(h) = k$ means it is player k's turn after history h,*
- *$(\mathcal{I}_i)$ is the information partition of $\{h \in H : P(h) = i\}$ for each player i with the property that $A(h) = A(h')$ whenever h and h' are in the same partition,*
- *u_i is the payoff function of player i associated with every terminal history.*

An extensive game with imperfect information can be described by a game tree, as shown in Figure 2. In the game tree, each node represents a player i; nodes connected by the dotted line are in the same information partition; the paths from the root to the nodes is the set of history H; and the payoff is shown after every terminal node. For example, the meaning of the leftmost path is: if player 2 takes strategy y_1 after player 1 takes strategy x_1, they both get payoff 1.

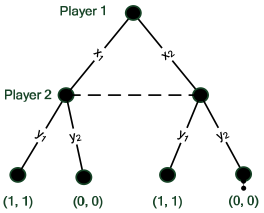


Figure 1: An example of an extensive game with imperfect information in the form of game tree.

3.2 Solution Concepts

A solution concept is a formal rule representing the game model's prediction, indicating how players in the game perform their actions. The most commonly-used solution concepts are equilibrium concepts. In this section, we give a formal definition of the Nash equilibrium and the sequential equilibrium.

Definition 3. **(Nash equilibrium)** *The Nash equilibrium of a normal-form game $(N, (A_i)_{i\in N}, (u_i)_{i\in N})$ is an action profile $a^* \in (\times A_i)$ of actions with the property that for every player $i \in N$, we have*

$$u(a^*_{-i}, a^*_i) \geq u(a^*_{-i}, a_i) \forall a_i \in A_i.$$

The Nash equilibrium is a commonly used equilibrium, whereby no player can increase his payoff by changing his action when he believes no other players would change. In the definition, a is the combination of actions of all players and a_{-i} is the set of actions of all players except player i.

A sequential equilibrium is a refinement of the Nash equilibrium for extensive games with imperfect information. In addition to specifying a strategy for each player, a belief system for each player stating which history has occurred in each information partition is needed. The pair of a strategy profile and a belief system is called an assessment, defined as follows.

Definition 4. **(Assessment)** *An assessment in an extensive game $(N, H, P, (\mathcal{I}_i)_{i\in N}, (u_i))$ is a pair (β, μ), where β is a profile of behavioral strategies and μ is a function that assigns to every information set a probability measure on the set of histories in the information set.*

In the definition, β is the behavioral strategy where $\beta(h)(a)$ denotes the probability that action a will be taken after history h, and μ is the belief system, where $\mu(I)(h)$ denotes the player's belief that to history h is in partition I.

Definition 5. **(Sequential rationality)** *An assessment (β, μ) is sequential rational in information set I if the players get their optimal payoff by adopting the strategy specified in β given their subsequent beliefs and others' subsequent strategies, as specified in the assessment.*

Definition 6. **(Consistency)** *An assessment (β, μ) is consistent if there exists a sequence of completed mixed strategy profiles $\{\beta_k\}$ such that $lim_{k\to\infty}(\beta_k, \mu_k) = (\beta, \mu)$, where μ_k is derived from β_k using Bayes' rule.*

Definition 7. **(Sequential equilibrium)** *An assessment is a sequential equilibrium of an extensive game with perfect recall if it is sequential rational and consistent.*

4. SOCIAL VERIFICATION MODELS

In this paper, we model human computation as the process of finding the descriptions of questions, i.e. given a question q from the question space Q, we want to find the appropriate description d from the description space D. For example, in the image annotation problem, the questions Q are images, and the descriptions D are labeling words. In the problem of locating objects in images, Q is the set of object names, and D is the set of relevant image parts.

To guarantee the quality of the outputs produced by human computation, social verification is commonly used since it is inherent with two desired properties. First, it can encourage users to perform the desired computation by rewarding users whose output matches with other users'. Moreover, verifications guarantee a high probability of correct output under the condition that more people behave correctly than those that do not.

In this paper, we model two two-player social verification mechanisms: simultaneous verification and sequential verification. We give the formal definitions and analysis in the following section.

4.1 Simultaneous Verification

4.1.1 *Definition*

Simultaneous verification describes the structure whereby users give descriptions of a question simultaneously. The descriptions are considered to be valid if two users agree on the same description. Formally, given the question $q \in Q$, each player $i \in \{1, 2\}$ can choose a description $d_i \in D$. When the two descriptions d_1 and d_2 match, users are rewarded and the system produces an output.

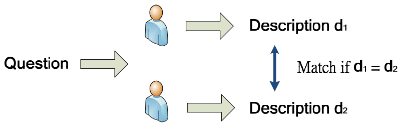


Figure 2: Simultaneous verification mechanism in human computation.

4.1.2 *Game theoretic modeling*

In a simultaneous-verification game, each player makes his decision without knowing the other player's decision. This kind of game can be defined as a normal-form game. In addition, simultaneous verification is characterized by three properties: (1) every player in the game is identical with respect to the game rules, i.e., changing the identity of the players would not change their payoffs; (2) both players get the same payoff; and (3) players get the optimal payoff when they perform the same actions. Based on these characteristics, simultaneous verification games can be modeled as a normal-form game.

Definition 8. **(Simultaneous verification game)** *Given a question q and a set of possible descriptions D, a human computation game with simultaneous verification can be modeled as a normal-form game, represented by a tuple $\Gamma = (N, (D_i)_{i \in N}, (u_i)_{i \in N})$, where $N = \{1, 2\}$, $D_1 = D_2 = D$ is the set of descriptions available for players, and (u_i) is the utility function of player i that satisfies the following three conditions:*

- $u_1 = u_2 = u$,
- $u(d_1, d_2) = u(d_2, d_1)$,
- $u(d_i, d_i) > u(d_i, d_j)$ *for all* $i \neq j$.

4.1.3 *Equilibrium analysis*

The game models in which players receive an optimal payoff when they choose the same strategy are called *coordination games*. The equilibrium analysis is straightforward.

Theorem 1. *A simultaneous-verification game has $n = |D|$ pure Nash equilibria, represented by the set S where*

$$S = \{(d, d) : d \in D\}.$$

The proof of the Nash equilibrium analysis is trivial since a rational player would not change his strategy if he takes the same strategy as the other player. However, the result of equilibrium analysis does not provide much useful information. We still do not know how a rational player would act since there are n equilibria that players can choose.

		Player 1		
		a_1	a_2	a_3
Player 2	a_1	(1,1)	(0,0)	(0,0)
	a_2	(0,0)	(1,1)	(0,0)
	a_3	(0,0)	(0,0)	(1,1)

Table 2: An example of a simultaneous verification game. The grey region marks the Nash equilibria of the game.

In traditional game theoretic analysis, the optimal strategy of the coordination game is a mixed strategy that randomly chooses one of the strategies in the n pure Nash equilibria. However, as the ESP Game shows, people can do much better than randomly choose their actions. Players in a game may coordinate to choose some preferred equilibria, which are called the focal points (a.k.a Schelling points).

Originally introduced by Schelling [11], focal points are the "prominent" or "salient" solutions to the game. For example, consider two players who are both required to choose an element from the set Z, and they only get a payoff when they choose the same element. Suppose each player differentiates the elements in terms of frequency, which means "how many times he/she has heard them mentioned." For instance, if each element is a single word, the frequency of the element is the times players have seen them in books and newspapers for some period of time. A natural focal point would be to choose the element with the highest frequency [12]. Intuitively, players would choose the element with higher frequency since the element is more "prominent" than others.

Returning to the simultaneous verification game, players try to choose an element d from the description set D given the question q. For simplicity, assume that all players have the same private description of the frequency function $f(d|q)$, which denotes how many times d appears when q is given. In this case, a natural focal point would be to choose the description d maximizing $f(d|q)$. Since the frequency is the statistic of the times of occurrence in a certain period, we will rewrite it in the form of probability $p(d|q)$ by normalization. Following the formula of conditional probability,

$$p(d|q) = \frac{p(d, q)}{p(q)} = \frac{p(d) \times p(q) \times r(d, q)}{p(q)} = p(d) \times r(d, q),$$

where $p(d, q)$ is the joint probability that elements d and q will co-occur, and $r(d, q) = \frac{p(d,q)}{p(d) \times p(q)}$ is defined to represent the relevance of the description d to the question q. When d and q are irrelevant, i.e., totally independent, $r(d|q) = 1$. Since the focal point of the game is to maximize $f(d|q)$, i.e., $p(d|q)$, we can conduct the following lemma.

Lemma 1. *In a human computation game with simultaneous verification $\Gamma = (N, (D_i)_{i \in N}, (u_i)_{i \in N})$, where each* *player's payoff is equal in every Nash equilibrium, players would choose description $d \in D$ satisfying*

$$d = argmax_{d \in D} p(d) \times r(d, q),$$

where $p(d)$ is the frequency of description d, and $r(d, q)$ is the relevance function indicating the relationship between description d and question q.

Intuitively, the player strategy in the focal points would be to choose the description d that maximizes the frequency of d and the relevance of description d to question q.

4.1.4 Analysis of human computation games with simultaneous verifications

In this section, we review some human computation games with simultaneous verifications and show how to apply our model and analysis to these games.

- The ESP Game. The typical example of simultaneous verification game is the ESP Game [2], in which the question space D is the set of images, the description space D is the set of labeling words, and the match condition occurs when two provided labels, d_1 and d_2, are the same. According to the analysis, player would choose the description d which maximizes $p(d) \times r(d, q)$, where $r(d, q)$ is the relevance relation between the image with taboo words q and the label d, and $p(d)$ is the properties of the label. This explains why players would tend to guess easier words, which maximize $p(d)$, and related words, which maximize the relevance relation $r(d, q)$ between the taboo word and the labels.

- Matchin [13] and Squigl[2]. In Matchin, Q is the set of image pairs, and $D = \{left, right\}$ indicates the preference of images. The player would choose the description $d \in D$ which maximize $p(d) \times r(q, d)$, where $p(d)$ is the frequency of choosing right or left, and $r(q, d)$ is the value of how preferred it is to choose the preference d given the image-pair q. Assuming players have the same frequency value in choosing right or left, Matchin is able to obtain the true preference relations between the image pair since players would maximize $r(q, d)$, i.e., choose the preference. In Squigl, Q is the set of image-word pairs, and D is the set of possible traces. Following the same analysis, players would choose to draw a rough sketch, i.e., maximizing $p(d)$, around the specified object, i.e., maximizing $r(q, d)$.

- Voting and PhotoSlap. The traditional majority-based (voting) mechanism can also be modeled as a simultaneous verification game. A n-player voting game can be regarded as C_2^n two-player simultaneous verification games, and the descriptions with the highest number of matchings are treated as the output. Moreover, users in a simultaneous verification game do not have to describe the question at the same time. In PhotoSlap [8], users are required to answer if two face photos are of the same person in two different stages of the game. Though the players do not answer the question together, PhotoSlap still can be modeled as a simultaneous verification because player performs their actions without knowing what the other players are doing.

4.2 Sequential Verification

4.2.1 Definition

The structure of sequential verification is similar to the popular game "Charade". The process can be split into two phases: the description phase and the guessing phase. In the description phase, the first player is required to provide descriptions to the given question. In the guessing phase, the second player tries to guess the original question given the descriptions. If the second player guesses correctly, both players will be rewarded and the system treats the descriptions as the output. Formally, given $q_1 \in Q$, player 1 can choose description $d \in D$, and player 2 can choose $q_2 \in Q$, which is his guess about the original question. Players are rewarded and system produces output when q_1 and q_2 match.

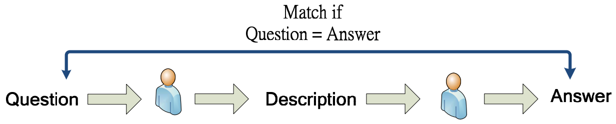


Figure 4: Sequential verification mechanisms in human computation.

4.2.2 Game theoretic modeling

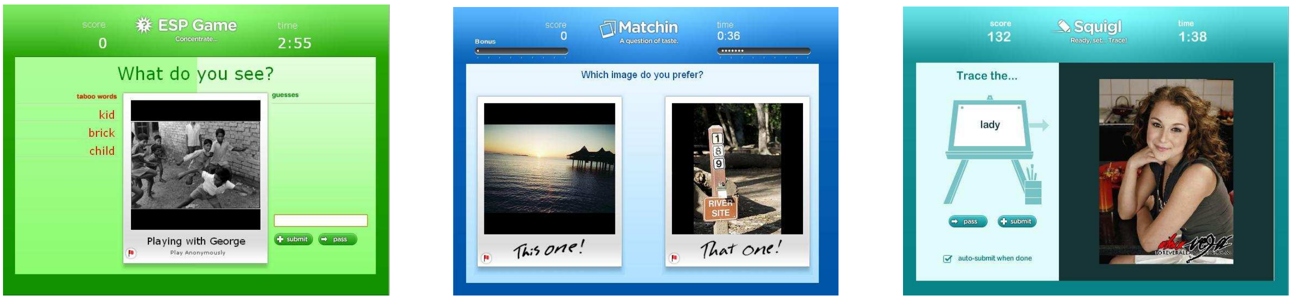
Besides the two players described above, we consider the system as a third player. First, the system player chooses a question q_1 from the question space Q. Then, player 1 chooses the description d from the description space D. Finally, player 2, who is not aware of question q_1, makes his guess of the original question q_2 from the question space Q. If the guess q_2 matches q_1, then both players are rewarded. This process can be modeled as a three-player extensive game with imperfect information.

Definition 9. **(Sequential verification game)** *Given a question space Q and a description space D, a sequential-verification game can be modeled as a three-player extensive game with imperfect information, represented by a tuple $\Gamma = (N, H, P, (\mathcal{I}_i)_{i \in N}, (u_i)_{i \in N})$, where*

- $N = \{system, 1, 2\}$, *and*
- *for all $q \in Q$, $d \in D$:*
  - $H = \{\phi\} \cup \{q\} \cup \{(q, d)\}$,
  - $A(\phi) \in Q$, $A(q) \in D$, *and* $A((q, d)) \in Q$,
  - $P(\phi) = system$, $P(q) = 1$, *and* $P((q, d)) = 2$,
  - $\mathcal{I}_2 = \{I(d_i) : d_i \in D\}$ *where* $I(d_i) = \{(q_j, d_i) : q_j \in Q\}$,
  - $u_1(q_i, d, q_j) = u_2(q_i, d, q_j) = \delta_{ij}$,

where δ_{ij} equals to 1 if $i = j$, and 0 otherwise.

[2] http://www.gwap.com/squigl-a



(a) The ESP Game. Players are given the image and required to give annotation words.

(b) Matchin. Players are given a pair of images and required to answer which one the partner prefers.

(c) Squigl. Players are given a word-image pair and required to give the locations of the word in the image.

Figure 3: Human computation games with simultaneous verifications.

By limiting the size of the question space Q and the size of the description space D to 2, the sequential verification game can be represented in the form of game tree, as shown in Figure 5. For the simplicity, we omit the payoff for the system player.

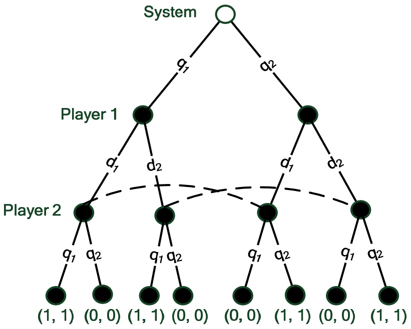


Figure 5: An example of sequential verification modeling in human computation. The payoff for the system player is omitted.

4.2.3 Equilibrium analysis

Similar to the analysis in simultaneous-verification game, we define a frequency function f in the following sequential equilibrium analysis. Given question space Q and the description space D, $f(q|d)_{q\in Q,d\in D}$ denotes the private description of the players on "how many times question q appears when given description d". We also rewrite it in the form of probability $p(q|d)$.

Players get optimal payoffs when player 2 guesses correctly about the original question q given description d. The behavior of player 2 is similar to that of players in simultaneous verification games, i.e. player 2 would guess the question q maximizing $p(q|d)$. However, the information we really care about is the behavior of player 1, since it directly relates to the system outputs. Under the belief that player 2 chooses question q maximizing $p(q|d)$, player 1 would choose the description d, such that $p(q|d) \geq p(q_i|d)$ for all $q_i \in Q$, i.e., q is the most frequent element co-occurring with d than any other questions. Player 2 would guess correctly if both players adopt these strategies. We write these strategies in the form of sequential equilibrium in the following lemma.

Lemma 2. *Given a question space Q, a description space D, and a frequency function f shared by all players, an assessment (β, μ) is a sequential equilibrium of the sequential-verification game $(N, H, P, (\mathcal{I}_i)_{i\in N}, (u_i)_{i\in N})$ if the following condition holds for all $q_i \in Q$:*

- $\beta_{system}(\phi)(q_i) = \frac{1}{|q|}$,
- $\beta_2(I(d))(q_i) = \begin{cases} 1, & \textit{if } q_i = argmax_{q\in Q}p(q|d) \\ 0, & \textit{otherwise.} \end{cases}$
- $\beta_1(q_i)(d) = \begin{cases} 1/|D_{q_i}|, & \textit{if } d \in D_{q_i} \\ 0, & \textit{otherwise.} \end{cases}$
- $\mu_2(I(d))(q_i, d) = \begin{cases} 1/|D_{q_i}|, & \textit{if } d \in D_{q_i} \\ 0, & \textit{otherwise.} \end{cases}$

$\beta_i(I)(d)$ is the probability to choose description d for player i in information set I, and D_{q_i} is the set of d which satisfies $q_i = argmax_{q\in Q}p(q|d)$ and $d \in D$. Intuitively, D_{q_i} is the set that contains all the descriptions that make question q "prominent".

Though the notations seem to be complicated, the results are intuitive. The result of the sequential equilibrium analysis shows, for rational players: (1) player 1 would describe the question q in the way that q is more frequently seen than other questions given the description d. While there are many possible descriptions satisfying the above condition, player 1 would choose one of them in random. (2) player 2 would choose the answer which best fits the description, i.e. which has the highest frequency. To prove the sequential equilibrium, the assessment should be examined if it satisfies sequential rationality and consistency. The proof for sequential rationality is trivial since both players cannot increase their payoff by changing the strategies. The consistency of the assessment can be proved by setting the elements of value 0 in β_1 and μ_2 to ϵ and normalizing the value of the other elements.

4.2.4 Analysis of human computation games with sequential verifications

Peekaboom [4], a web game for locating objects in images, is an example demonstrating the sequential verification, where

Q is the set of the labels, D is the set of all possible combinations of image portions, and the match condition occurs when the original question q_1 equals to player 2's prediction q_2.

The music annotation game TagATune [6] is another example, which runs two sequential verification games simultaneously. In each sequential verification, Q is the set of songs, and D is the set of annotation words. However, instead of choosing $q_2 \in Q$, player 2 is given a hint question (song) q and only need to choose whether the question song is the same as the hint song. (q or not q). The players are rewarded and system produces a output when the match condition occurs in both sequential verification games.

According to the equilibrium analysis, player 1 would randomly choose one of the descriptions that are able to make the question "prominent", i.e., the chosen description d should satisfy $p(q|d) \geq p(q^*|d)\ \forall q^* \in Q$ given the question q. While this strategy is more descriptive than the strategy taken in the simultaneous verification game, there are more uncertainties on the descriptions player may choose. For example, when given the term "elephant" and an image containing an elephant in Peekaboom, player could choose the description that reveals (1) the image parts of the whole elephant, (2) the image parts of the nose, (3) the image parts of the elephant head, or even (4) the image parts of a portion of the elephant head. The situation is similar in TagATune, in which player may describe a song in various way. These results need further analysis in order to represent useful information.

5. EVALUATION

In the previous section, we give a game theoretic analysis of two social verification mechanisms and demonstrate how to apply the modeling and analysis in several human computation games. While the preliminary results support our analysis, we do not compare the system output of two mechanisms directly since the target problems and game designs are not the same.

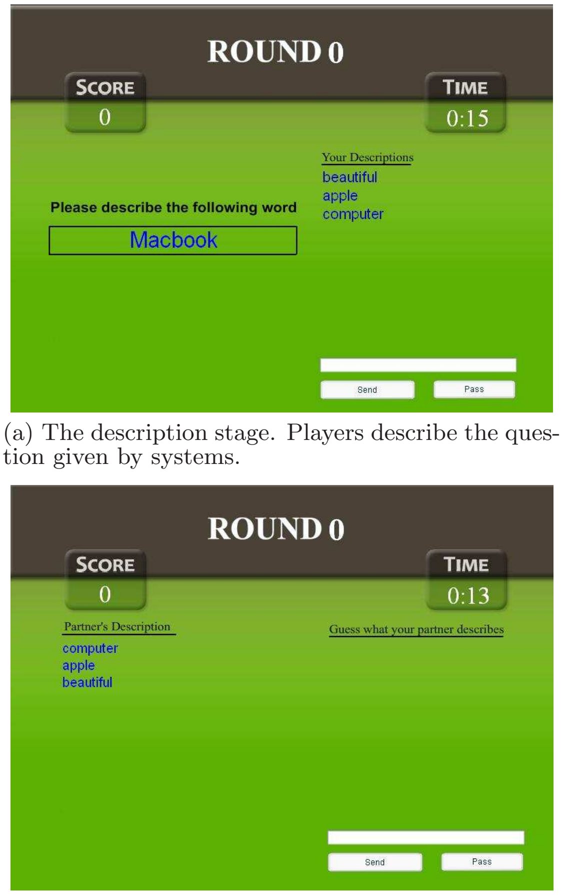
Therefore, we conduct experiments on a simple scenario: give the descriptions for the specified terms. The question space Q and the description space D are both the set of words. Given the question space Q to be a set of 120 words, we apply two proposed social verification mechanisms in this scenario and collect their system outputs.

5.1 Experiment Setup

There are two stages in this task: the description stage and the guessing stage, as shown in Figure 6. In the description stage, players are given a word and are required to give descriptions or any related terms. In the guessing stage, players are given some descriptions and are required to guess which word is being described.

This human intelligence task is implemented in two mechanisms in the form of games and published on Amazon Mechanical Turk to collect user input.

- Simultaneous verification game (SIM Game): This is a two-player process. Similar to the ESP Game, players are given the same term in the description stage. In each round, they have 30 seconds to match with each other. Players get paid if they finish the game and match descriptions at least 10 rounds in the 15 rounds of playing. Descriptions that are matched by players are recorded.
- Sequential verification game (SEQ Game): This is a two-player process. In each round, one player is in the description stage to describe the given term, and the other is in the guessing stage to guess the term. Players win that round if the guessing player guess correctly in 30 seconds. Players get paid if they finish the game and win at least 10 round in the 15 rounds of playing.



(a) The description stage. Players describe the question given by systems.

(b) The guessing stage. Players guess the original question according to the provided descriptions.

Figure 6: Snapshots of the evaluations conducted on Amazon Mechanical Turk.

5.2 Experiment Results

The simultaneous verification game (SIM Game) has been played for 154 times, whereby 1878 labels are generated, and the sequential verification game (SEQ Game) has been played for 178 times, in which 4368 labels are generated.

In the 1878 labels collected by SIM Game, there are 384 distinct labels, and there are 2634 distinct labels out of 4368 total labels in SEQ Game. The result demonstrates that SEQ Game leads to a more diverse output than SIM Game.

Moreover, the labels in SEQ Game is usually more descriptive than that in SIM Game, as demonstrated in Table 3. On average, each label contains 2.633 words in SEQ Game, and 1.002 words in SIM Game. While more words represents for more descriptive powers, we can implicitly infer that labels in SEQ Game is more descriptive.

Another interesting discovery is the ratio of the mis-spelling words. There are 335 mis-spelling words in the 4368 labels of SEQ Game, and no mis-spelling labels in SIM Game. This result may be dure to the stronger verification in ensuring the correctness in SIM Game.

SIM Game	SEQ Game
add	science with numbers
class	adding, subtracting
calculus	subject in calculations
	school course with numbers

Table 3: Labels collected for "Mathematics."

5.3 Discussion

This small-scale experiments have revealed some important properties of the labels generated in two social verification mechanisms. First, labels collected in sequential verification mechanism are more descriptive and diverse than the labels in the simultaneous verification mechanism. The result confirms the game theoretic analysis of player behavior in the previous section.

Understanding player behavior and system outputs in different social verification mechanisms is essential in designing human computation applications. For example, adopting sequential verification mechanism might be a better choice when description diversity is important in the application. However, additional mechanisms, such as spell checking, should be applied to ensure the data quality. If the description space is limited, e.g., the two choices in Matchin, or the output correctness is essential, simultaneous verification mechanism would be a better fit.

6. CONCLUSIONS

In this paper, we propose formal models for two fundamental social verification mechanisms, simultaneous verification and sequential verification, in a game theoretic approach. Our contributions are summarized below:

- We have shown that social verification mechanisms in human computation games can be modeled by using game theory. Since simultaneous and sequential verifications are fundamental, our models can be generally applied to human computation systems.
- We have shown that simultaneous and sequential verifications can lead to different system outcomes. While sequential verification promotes a more diverse and descriptive labeling, simultaneous verification is stronger in ensuring the data correctness.
- Real-world experiments involving two mechanisms have been conducted. Through analyzing the data generated by the different mechanisms, we summarize the properties of the system output and give suggestions for choosing mechanisms when designing new human computation applications.

In our future work, we will identify other verification mechanisms and explore the possibility of integrating competitive elements into human computation. In addition, we will investigate other issues from a game theoretic perspective, including the presence of malicious players and how to change the incentive structure to achieve different system outcomes.

7. REFERENCES

[1] Luis von Ahn. Games with a purpose. *IEEE Computer Magazine*, 39(6):92–94, 2006.

[2] Luis von Ahn and Laura Dabbish. Labeling images with a computer game. In *CHI '04: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 319–326. ACM Press, 2004.

[3] Ingmar Weber, Stephen Robertson, and Milan Vojnovic. Rethinking the ESP game. Technical report, Microsoft Research, September 2008.

[4] Luis von Ahn, Ruoran Liu, and Manuel Blum. Peekaboom: a game for locating objects in images. In *CHI '06: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 55–64. ACM Press, 2006.

[5] Luis von Ahn, Mihir Kedia, and Manuel Blum. Verbosity: a game for collecting common-sense facts. In *CHI '06: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 75–78. ACM Press, 2006.

[6] Edith L. M. Law, Luis von Ahn, Roger B. Dannenberg, and Mike Crawford. Tagatune: A game for music and sound annotation. In *International Conference on Music Information Retrieval (ISMIR' 07)*, pages 361–364, 2003.

[7] Luis von Ahn and Laura Dabbish. Designing games with a purpose. *Commun. ACM*, 51(8):58–67, 2008.

[8] Chien-Ju Ho, Tsung-Hsiang Chang, and Jane Yung-jen Hsu. Photoslap: A multi-player online game for semantic annotation. In *Twenty-Second Conference on Artificial Intelligence (AAAI-07)*. AAAI Press, July 2007.

[9] Shaili Jain and David C. Parkes. A game-theoretic analysis of games with a purpose. In *Internet and Network Economics, 4th International Workshop (WINE 2008)*, pages 342–350, December 2008.

[10] Martin J. Osborne and Ariel Rubinstein. *A course in game theory.* The MIT Press, Cambridge, Massachusetts, 1994.

[11] Thomas C. Schelling. *The Strategy of Conflict.* Harvard University Press, Cambridge, Massachusetts, 1960.

[12] Robert Sugden. A theory of focal points. *The Economic Journal*, 105(430):533–550, 1995.

[13] Severin Hacker and Luis von Ahn. Matchin: Eliciting user preferences with an online game. In *CHI '09: Proceedings of the SIGCHI conference on Human factors in computing systems.* ACM Press, 2009.