# Turing Trade:
# A hybrid of a Turing test and a prediction market

Joseph Farfel and Vincent Conitzer

Duke University

**Abstract.** We present Turing Trade, a web-based game that is a hybrid of a Turing test and a prediction market. In this game, there is a mystery conversation partner, the "target," who is trying to appear human, but may in reality be either a human or a bot. There are multiple judges (or "bettors"), who interrogate the target in order to assess whether it is a human or a bot. Throughout the interrogation, each bettor bets on the nature of the target by buying or selling human (or bot) securities, which pay out if the target is a human (bot). The resulting market price represents the bettors' aggregate belief that the target is a human. This game offers multiple advantages over standard variants of the Turing test. Most significantly, our game gathers much more fine-grained data, since we obtain not only the judges' final assessment of the target's humanity, but rather the entire progression of their aggregate belief over time. This gives us the precise moments in conversations where the target's response caused a significant shift in the aggregate belief, indicating that the response was decidedly human or unhuman. An additional benefit is that (we believe) the game is more enjoyable to participants than a standard Turing test. This is important because otherwise, we will fail to collect significant amounts of data. In this paper, we describe in detail how Turing Trade works, exhibit some example logs, and analyze how well Turing Trade functions as a prediction market by studying the calibration and sharpness of its forecasts (from real user data).

**Key words:** prediction markets, Turing tests, games with a purpose, deployed web-based applications, using points as an artificial currency

## 1 Introduction

In a Turing test, a single human being (the *judge*) chats with two mysterious conversation partners [6]. One of the two mystery conversationalists is another human, while the other is a computer program (a *chat bot*, or just *bot*). The bot is the entity who is actually taking the test: If the judge cannot (accurately) tell which mystery conversation partner is the human and which is the bot, then the bot passes the test (and otherwise it fails). It is easy to see that a Turing test can also be run with only a single mysterious conversation partner (whom we will call the *target*). To do so, the test organizer chooses a human target with probability 50% and a computer target with probability 50%. Then, after a conversation with the target, the judge is asked to report how probable she thinks it is that the target is a human—if she reports 50% or higher when talking to a bot, then that bot passes the test.

One might imagine a variant of the Turing test where the "judge" consists of a *group* of humans (more generally, *agents*), and the result of the test hinges upon the group's aggregate belief of the probability that the target is a human. This setting involves multiple judging agents, where each agent has her own belief, but where information about beliefs might be exchanged among agents during the course of the test. An agent's personal belief is updated throughout the test based on the information she receives about other agents' beliefs, and, of course, on the target's contributions to the conversation.

Our new web game, Turing Trade, is an implementation of a Turing test with a group as the judge. In Turing Trade, a group of agents converses with a single target. Each individual agent in the group gets to ask the target public questions, and the target gives public answers. During the conversation, all individuals in the group are encouraged to competitively bet on the target's humanity, by buying and selling securities (with points, not real money). The price of these securities varies based on judges' bets, and at any given time in the game, this price is a measure of the group's consensus belief that the target is a human. At the end of the game, the target's true nature (human or computer) is revealed, and based on this some of the securities pay out. The betting part of the game is a *prediction market* [9], where the single binary event that the judges are trying to predict is "the target will be revealed to be a human."

Turing Trade can be played online at http://www.turingtrade.org. All logs from played games are posted publicly on the website. There are previously existing websites where one can take a more traditional Turing test, notably the Turing Hub, at http://www.turinghub.com, where a single player can log in as a judge, have a conversation with a target, and then rate the target's humanity on a four-point scale. One goal of Turing test web sites is to gather data from humans to help improve the conversation skills of bots. Having a large database of conversation logs, each with some attached humanity rating, would certainly be valuable for designing and training chat bots, and possibly for AI in general. We believe that Turing Trade has at least the following advantages over more traditional Turing test websites (such as the Turing Hub):

1. **Entertainment.** We believe that playing Turing Trade is more fun than participating in a normal Turing test. Apart from the social amusement provided by the interesting and clever questions submitted by other members of the judging group, the game encourages competition, by rewarding judges who increase the accuracy of the consensus probability estimate.
2. **More data, from more volunteer judges.** *Games with a purpose* use entertainment value to convince legions of humans to do something useful that is (currently) difficult for computer programs [8]. For example, playing the ESP Game, at http://www.espgame.org, is a fun way to help put useful labels on all of the images on the web [7]. In the first four months of the ESP Game's existence, 13,630 people played the game (over 80% of which played on more than one occasion), and an informal recent check of the website at various times of day implies that about 40 people are playing the game at any given moment. Very few people (certainly, fewer than the numbers mentioned) would sit and label images without compensation if it were not in the context of a game (with competition, cooperation, points, etc.). Similarly, Turing Trade's goal is to attract more Turing test judges (and human targets) than its non-game contemporaries.

3. **Better data, through proper incentives.** Most Turing test web sites offer no incentives to the judge. Even if the mere act of having a conversation with a mysterious subject is incentive enough for people to participate in the test, a judge is certainly not strictly incentivized to report her truthful belief about the nature of the target at the end. Turing Trade's prediction market betting system incentivizes a bettor to bet in a way reflecting the true probability she assigns to the target being a human; moreover, it encourages the bettor to improve her own acuity at estimating this probability, by punishing those who predict incorrectly, and rewarding those who predict correctly. (Punishments and rewards are in the form of points, rather than real money, but this is better than no incentive at all, and in fact the lack of real money does not seem to greatly affect the accuracy of a prediction market [5].)
4. **More mystery.** Judges having conversations at the Turing Hub are immediately biased toward thinking that they are speaking to a bot: since the site has only light traffic, the chances of a human-human conversation are quite low, and to make things worse, some of the bots on the site use custom (and very bot-like) message windows. More player traffic (combined with the ability to play as a target), as well as a consistent interface, causes Turing Trade's targets to be more mysterious (which is also more enjoyable).
5. **Fine-grained data.** In Turing Trade, a group's current consensus evaluation of the probability that the target is a human is given by the current price of the securities. Since this price varies over the course of a conversation, our data not only gives an overall assessment of how human-like a target acted in a particular conversation, but also shows how the impression that the target made varied over time. For example, a game log might show that the security price stayed high for a while, and then dropped sharply after the target answered question 5. This would imply that the target gave human-like answers to questions 1-4, but not to question 5. One can imagine mining mountains of logs for sharp price drops and rises, thereby compiling lists of good questions, as well as good and bad answers to them. This should help in the design of better chat bots as well as in the training of judges. We provide some examples of log data generated by Turing Trade in Section 4.

Apart from web-based Turing tests like those at the Turing Hub, there are a few regular Turing test-based competitions, some offering cash prizes to the most human-like participating bot. The most famous of these is the Loebner Prize, which claims to be the first formal instantiation of a Turing test (http://www.loebner.net/Prizef/loebner-prize.html). This yearly competition, started in 1990, features four judges (usually university professors), each of whom scores every entering bot. The Loebner Prize offers a $100,000 grand prize and a solid gold medal to the first bot whose responses are "indistinguishable from a human's." Although this prize goes unclaimed, an annual prize of $2,000 is offered to the most human-like bot in the competition.

Though they work fine as Turing tests, and are good indicators of which bots are currently the most advanced, the Loebner Prize competition and other competitions like it do not serve the same purpose as Turing Trade. Turing Trade's purposes include: (1) to collect large quantities of fine-grained data for use by bot designers, (2) to introduce a novel, fast-paced prediction market, which may provide valuable lessons for the design of other prediction markets, and (3) to provide entertainment value.

## 2 Game Overview

In a game of Turing Trade, the *target* is the single player (possibly a bot) whose humanity is being judged. The group judging the target is composed of $n$ agents, called *bettors*. The bettors ask the target questions, and bet on whether the target is a human or a computer. The target answers questions from the bettors, and tries to seem as human as possible, whether or not it is really a human.

As a brief note on implementation, all players in the game (bettors and target) communicate through web-based Java applets. These applets send all information through a central server, also written in Java. The server is capable of managing multiple simultaneous Turing Trade games. In the current incarnation of the game, the number of bettors, $n$, is restricted to three, at most, per game (this is not due to scalability reasons but rather to ensure that all bettors have a chance to ask questions).

### 2.1 Bot Targets

It is very important for our game to have a strong lineup of bots available to serve as targets. The bots described below (except for Simple Bot) were written by others, and reside on their owners' web servers (it is not our intention to create new bots ourselves). When a game is in progress, the Turing Trade server initiates a new conversation with a bot, and simply sends it bettors' questions and receives the bot's answers. The current incarnation of the game features six different bots organized into three classes:

1. **Simple Bot.** This is a very simple bot—to any question, it replies with the same answer ("Hmmm... That's an interesting question.").
2. **Alice and iGod.** These bots are based on AIML, or the Artificial Intelligence Markup Language. AIML and Alice, the first bot to use it, are creations of Dr. Richard Wallace (http://www.alicebot.org); they are extensions of the logic underlying the classic bot Eliza, developed by Joseph Weizenbaum in 1966. The iGod bot is currently the most popular bot at the free AIML-bot hosting web site Pandorabots (http://www.pandorabots.com/). Alice won the Loebner Prize for most human-like chat bot in 2000, 2001, and 2004.
3. **Jabberwacky, George, and Joan.** These three bots are all based on Jabberwacky, by Rollo Carpenter (http://www.jabberwacky.com/). Its approach is heavily centered on learning, and it operates primarily by storing everything that any human has ever said to it, and using contextual pattern matching to find things in this vast database to say in response to new human input. Given this emphasis on learning, Jabberwacky-based bots are especially good candidates for using Turing Trade logs (which include not only a conversation log, but also information about the varying perception of the target's humanity during the conversation) to improve performance. The Jabberwacky bots George and Joan won the Loebner prize in 2005 and 2006.

### 2.2 Questions and Answers

The target always sees only one question at a time from the group of bettors. This question is called the *current* question. The target considers the current question, and

sends its answer to the server; at this point, the server may send the target another single (current) question. The target may only send one answer for each question. From the target's perspective, the conversation is a simple back-and-forth exchange. The only indication the target has that it is talking to a group of people rather than a single person is that questions belonging to bettor $i$ are tagged as such. This allows chat bots that currently exist to play Turing Trade unmodified.

Every bettor also gets to see the current question, at the same time as the target. Unlike the target, however, which may only submit an answer if there is an unanswered current question, any of the $n$ bettors may submit a question to the game server at any point during the game. The server keeps a queue of questions, $Q_i$, for each player $i$, and initially, all question queues are empty. When the server receives a question, $q$, from a bettor $i$, it does the following:

- If there is no current (unanswered) question, broadcast question $q$ to all bettors and the target. Question $q$ is now the current question.
- Otherwise (there is a current question), add the question $q$ to queue $Q_i$.

With this setup, all bettors and the target see the current question, but every other unanswered question is invisible to everyone but the bettor who asked the question. When the server receives an answer from the target to a current question $q$, it does the following:

1. Let $i$ be the bettor who asked the question being answered (question $q$). Send the answer to bettor $i$, and send a signal (not containing the text of the answer) to every bettor $b \neq i$ signifying only that an answer to the question has been given.
2. Starting with $j = i + 1 \pmod n$, and incrementing $j \pmod n$ after each check, search for the first nonempty queue $Q_x$.
   - If all queues are empty, do nothing (except wait for a bettor to send a question).
   - Otherwise ($Q_x$ is the first nonempty queue), remove the first question $q'$ from $Q_x$. This is the new current question; send $q'$ to all bettors and the target.
3. Five seconds after step 1 (sending the answer to question $q$ to player $i$), send the text of the answer (to $q$) to every bettor $b \neq i$.

This scheme ensures that questions are taken from bettors in a round-robin manner, unless some bettors are not asking questions, in which case they are skipped. We note that bettor $i$ gets to see the answer to her question five seconds earlier than every bettor $b \neq i$. This delay rewards bettors for asking good questions (where a good question is one that reveals a lot about the nature of the target), because it allows the bettor who asked the question to trade on this information before it becomes available to the other bettors. An example of a bettor's view of an in-game conversation is shown in Figure 1.

### 2.3 The End of the Game

A game of Turing Trade ends in one of several ways:

- **Time runs out.** Each game is timed, and the time limit is fairly short (in the current incarnation, it is two minutes). While Alan Turing hypothesized that machines in the

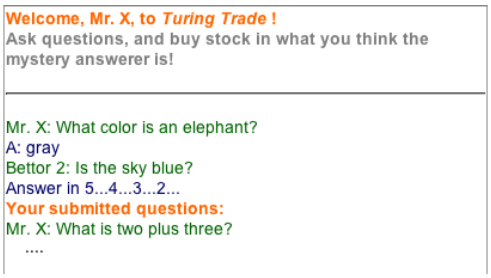



**Fig. 1.** An example of a bettor's ("Mr. X") in-game view of a conversation between the bettors and the target. This bettor owns the current question, shown at the bottom. She is also about to see the answer to Bettor 2's question, which Bettor 2 saw four seconds ago.

year 2000 with 119 MB of memory would be able to regularly pass a five-minute Turing test [6], this prediction certainly did not come to pass. Our empirical results show that bettors usually (but not always) become extremely certain of a target's nature even before the end of two minutes.

– **All bettors signify that they are done betting.** If a bot (or human) gives a particularly elucidating answer or two, bettors may become virtually certain of the target's nature. We give bettors the option to signify that they are satisfied with their current bets, and wish to end the game early. All bettors must agree to end early.
– **The target (or all of the bettors) leaves the game early.**

At the end of a game, the nature of the target is revealed to all of the bettors. The bettors and the target are rewarded based on the bettors' bets and the target's true nature.

### 2.4 Betting

At any time during the conversation with the target, any bettor may place a bet on whether or not the target is human. A bet is made by buying or selling a *human security* or a *computer security*. A human security is an asset of the form "Pays 100 points if the target is revealed to be a human," while a computer security is an asset of the form "Pays 100 points if the target is revealed to be a computer." Securities pay out at the end of the game, when the target's nature is revealed to bettors: for example, if the target is a human, a human security pays out 100 and a computer security pays out 0. Since the two types of security are complementary (owning one of each type of security is equivalent to owning 100 points), we without loss of generality restrict every bettor to own at most one type of security at a time.

Human and computer securities are bought from and sold to a central *market maker*, who has an infinite supply of securities to sell, and an infinite willingness to buy securities. The market maker always sets the price for the computer security at 100 minus the price of the human security (this is ignoring a small bid-ask spread that we will discuss shortly). A bettor can purchase or sell one security at a time. When a human security is purchased (or a computer security is sold), the price for human securities increases by


**Fig. 2.** The interface a bettor uses to buy and sell human and computer securities. The pictures indicate "betting human" (which means either buying human securities, or selling computer securities), and "betting computer" (which means either selling human securities, or buying computer securities). The number of securities the bettor owns, and the securities' type, is shown below the buttons. If the security price reaches a steady (and boring) equilibrium (usually at 100 or 0), a bettor can click "done buying;" if all bettors do this, the game ends early.

1, and when a human security is sold (or a computer security is purchased) the price for human securitites decreases by 1.

The market maker maintains a spread of 1 between bid and ask prices. This is done to prevent arbitrage: with the spread, a bettor can buy a security for the ask price of $x$ from the market maker (causing the ask price to increase to $x + 1$), and then sell it back for the bid price $(x + 1) - 1 = x$. Neither the bettor nor the market maker profits if this happens, but without the spread, the bettor would have had a profit of 1. The maximum price for a human security is 100, and the minimum price is 0. An example of the interface that a bettor uses to buy and sell securities is shown in Figure 2.

The price to buy a human security is plotted over the course of a game (an example is shown in Figure 3). Local equilibria in the price measure the bettors' consensus belief (at some point in time) of how probable it is that the target will be revealed to be a human. For example, if the human security price is hovering around 70, then the bettors, in aggregate, believe that the there is a 70 percent chance that the target will be revealed to be a human at the end of the game. This interpretation relies upon the assumption that the bettors are rational (in the sense of maximizing their expected number of points), and upon the fact that a prediction market such as this one offers incentives for rational bettors to update the consensus probability in ways consistent with their true beliefs (at least for a myopic sense of rationality).

## 3 Evidence for the Accuracy of Prediction Markets

Empirical evidence has shown that prediction markets are quite good at forming accurate probability estimates for events. For example, the Iowa Electronic Markets outperformed 451 out of 591 major public opinion polls in predicting the margin of victory in past U.S. presidential elections [1]. Perhaps surprisingly, even markets using play money exhibit very strong predictive powers. Pennock *et al.* discovered high prediction accuracy both for the Foresight Exchange (http://www.ideosphere.com), where traders bet on the outcomes of open scientific questions, and for the Hollywood Stock

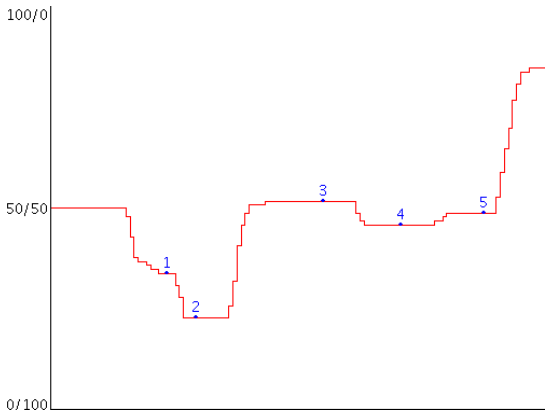



**Fig. 3.** An example of the price graph shown to the bettors and the target during a game of Turing Trade. Time progresses left to right, while the y-axis shows the price to purchase a human security. Dots indicate when the target answered a question. As one might expect, large shifts in the security price usually occur shortly after bettors see an answer (and update their impression of the target accordingly), while the price reaches equilibrium between answers.

Exchange (http://www.hsx.com), where ending security prices for Oscar, Emmy, and Grammy awards were found to correlate well with actual award frequencies [3]. Similarly, Servan-Schreiber *et al.* found no statistically significant difference in the accuracy of play money and real money prediction markets in predicting the outcomes of American Football games during the 2003-2004 NFL season [5]. Results like these bode well for the accuracy of Turing Trade's prediction market. This is especially promising because as bots improve their conversation skills, and become less distinguishable from humans, judges' predictions will need to become more accurate to detect what subtle differences remain. In Section 5, we assess the predictive powers of Turing Trade directly, based on real data.

## 4 Example Logs

Figures 4, 5, and 6 contain excerpts of some real logs from Turing Trade. The graph at the top of each log shows time on the x-axis, and the human security price on the y-axis. The log detailing the game times at which events happened appears below. Logs have been edited so that they show all questions and still fit within the space limits (this involved removing large chunks of entries detailing each new computer or human bet; some entries describing players joining or leaving the game have also been removed).

## 5 Calibration and Sharpness

How can we evaluate whether our prediction market is functioning well? One desirable property is that the predictions are *calibrated*. This means the following. Suppose we

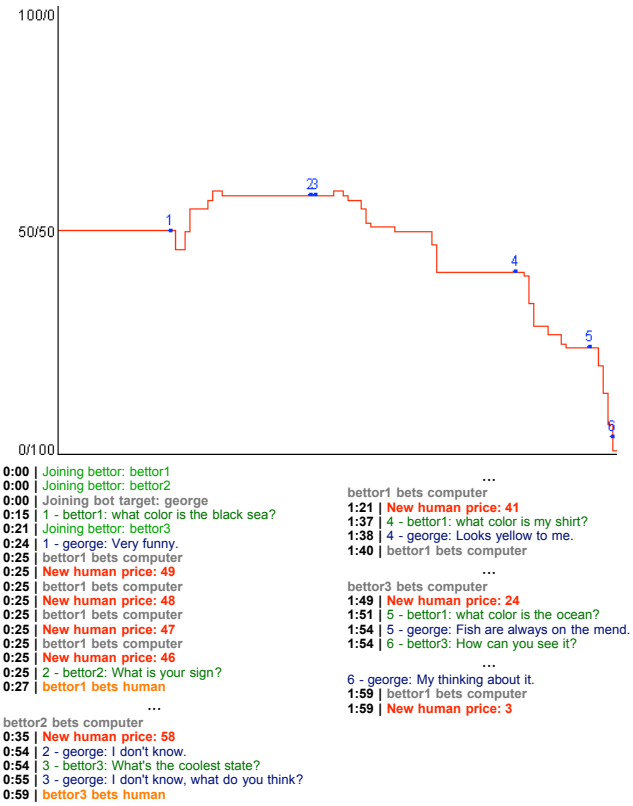
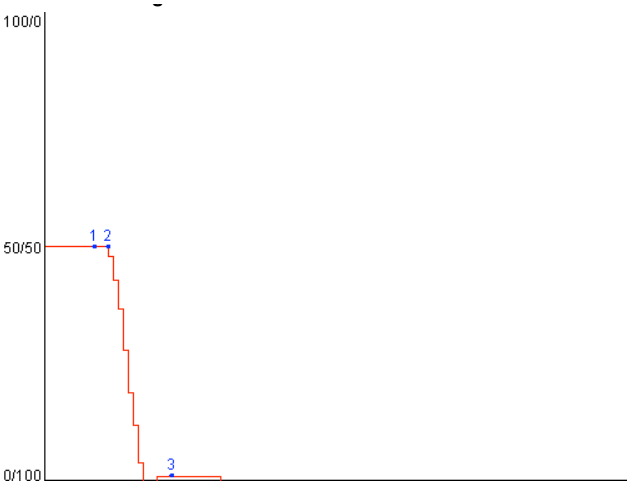



**Fig. 4.** This run demonstrates the ability of bots to sometimes evade conclusive detection for an extended period during a game. In this run, the Jabberwacky-based bot George was able to seem at least somewhat human, until its ridiculous answer to question 5.

consider all the runs where, after a given amount of time, the market probability (price) that the target is human is at (say) 10%. We would hope that in exactly 10% of these runs, the target is indeed a human. If this is true for all probabilities, then the market is (perfectly) calibrated.

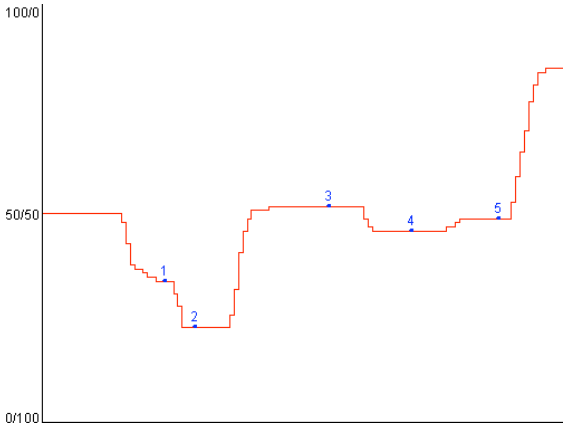
A practical problem with this definition is that we generally do not have many data for each individual probability. To address this, it is common to bin the probabilities together. For example, we consider all the runs where, after a given amount of time, the market probability of a human target is between 10% and 20%, and ideally the fraction of these runs where the target is indeed a human is between 10% and 20%. In practice, even this is often not the case for every bin, but we would hope that the market probabilities and the true fractions are at least close.

Currently, about 900 games of Turing Trade have been played. After removing the logs from games where no bets were made, we examined the remaining 694 game logs to determine market calibration. Figure 7 illustrates the results. The market seems reasonably, albeit not perfectly, calibrated. One would expect that the market would




**0:00** | Joining bettor: bettor1
**0:00** | Joining bettor: bettor2
**0:00** | Joining bettor: bettor3
**0:00** | **Joining bot target: Jabberwacky**
**0:09** | 1 - bettor3: Who won the basketball game?
**0:10** | 1 - Jabberwacky: The North.
**0:11** | 2 - bettor1: what's very blue?
**0:13** | 2 - Jabberwacky: Cake is very yummy.
**0:13** | **bettor3 bets computer**
**0:13** | **New human price: 49**

**Fig. 5.** Sometimes, a bot will seal its fate with its very first answer.




**0:00** | Joining bettor: bettor1
**0:00** | Joining bettor: bettor2
**0:00** | **Joining human target: target**
**0:08** | 1 - bettor1: hey there bot
**0:18** | **bettor1 bets computer**

...

**0:28** | 1 - target: are you talking to me?
**0:28** | 2 - bettor2: What kind of milk do you drink?
**0:30** | **bettor1 bets computer**
**0:30** | **New human price: 33**
**0:30** | **bettor1 bets computer**
**0:30** | **New human price: 32**

...

2 - target: whole milk
**0:43** | **bettor1 bets human**
**0:43** | **New human price: 24**

...

**0:56** | 3 - bettor2: Eww gross!
**1:06** | 3 - target: how's that?
**1:09** | 4 - bettor1: milk's the best
**1:14** | **bettor1 bets computer**

...

**1:25** | 4 - target: how about soy milk
**1:31** | Joining bettor: mquirk
**1:33** | **bettor2 bets human**

...

**1:39** | 5 - bettor1: this conversation isn't going anywhere
**1:45** | 5 - target: i agree
**1:48** | **bettor1 bets human**

...

**Fig. 6.** A human target. This log is an example of how a human can sometimes seem like a computer, even when his answers to questions are perfectly reasonable.

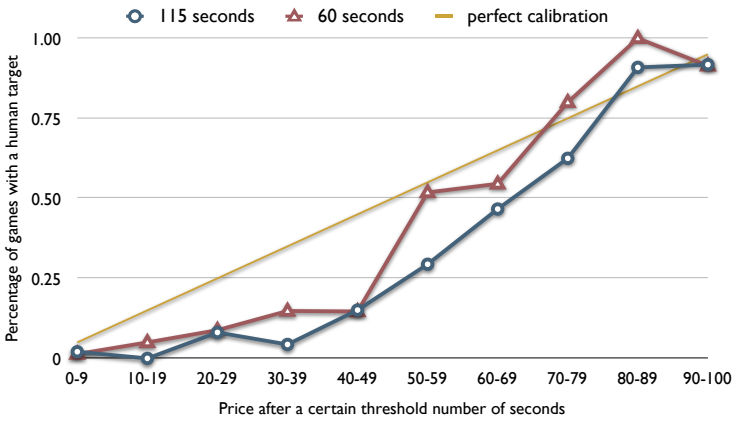



**Fig. 7.** The $x$-axis denotes bins of runs, partitioned by the human security price after a given number of seconds. The $y$-axis is the fraction of runs in a bin that had a human target. If the market were perfectly calibrated, these would match.

become even more calibrated over time, especially as players accrue more experience and become better bettors.

To have a good prediction market, it is not sufficient that it is calibrated. For example, suppose it is known that 50% of targets are human, and the initial market probability is always 50%. Then, if traders never trade at all, the prediction market is perfectly calibrated—but this would constitute a completely dysfunctional prediction market. The missing property is that of *sharpness*: we want the market predictions to be close to 0% or 100%. As it turns out, the Turing Trade prediction market makes very sharp predictions, as illustrated by Figure 8.

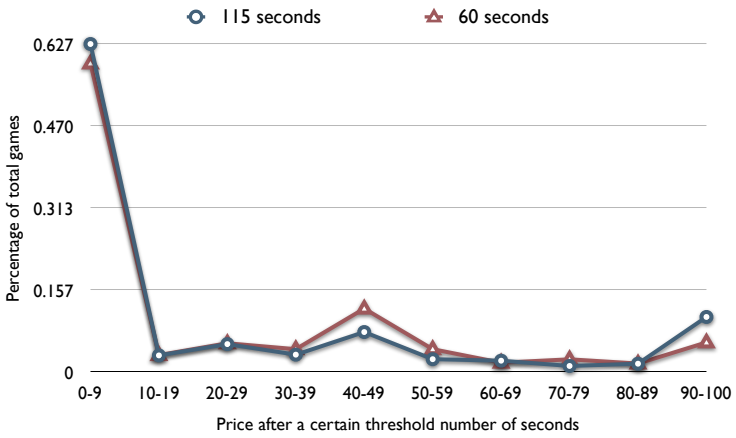



**Fig. 8.** The $x$-axis denotes bins of runs, partitioned by the human security price after a given number of seconds. The $y$-axis is the fraction of total runs in a bin. The high percentages at the lowest and highest bins indicate sharpness, while the spike at the $40-49$ bin is undesirable.

## 6 Conclusions

We introduced a new "game with a purpose" called Turing Trade. The game is a group-judged Turing test, where members of the judging group (bettors) bet on whether their mystery conversation partner (the target) is a human or a computer. Betting is accomplished through the use of a prediction market, with bettors using play money to buy and sell "human securities" and "computer securities" from an automated market maker. We believe the game offers numerous advantages over standard Turing test websites, including the promise of collecting significantly more data, and more finely-grained data, for chat bot designers and others. The Turing Trade project has additional purposes, including the creation of a novel, fast-paced prediction market that may provide useful lessons for the design of prediction markets in general. Another purpose is simply to create entertainment value for its players. If the game ends up being played by very many people, then, for example, providers of free e-mail accounts could use our game to ensure that bots do not sign up for accounts (a trick commonly used by spammers), by requiring a new user to play as the target in Turing Trade (and be judged human).

Prediction markets have worked very well, empirically, in other settings [1], even when fake money is used (as is the case with Turing Trade) [3, 5]. We have already obtained a significant number of runs with our publicly available web-based implementation of Turing Trade (http://turingtrade.org), especially after it received some attention on blogs including http://www.midasoracle.org and http://www.marginalrevolution.com. Our analysis of these runs suggests that Turing Trade produces very strong and quite accurate predictions after short periods of time, with the market price responding rapidly to good or bad answers by the target.

## Acknowledgments

We thank the National Science Foundation for support under award number IIS-0812113, the Alfred P. Sloan Foundation for support under a Research Fellowship, and Yahoo! for support under a Faculty Research Grant. We also thank all the people that have played Turing Trade and have given us valuable feedback.

## References

1. J. Berg, R. Forsythe, F. Nelson, and T. Rietz. Results from a Dozen Years of Election Futures Markets Research. *Handbook of Experimental Economics Results*, 2001.
2. R. Hanson. Logarithmic market scoring rules for modular combinatorial information aggregation. *Journal of Prediction Markets*, 1:3–15, February 2007.
3. D. M. Pennock, S. Lawrence, C. L. Giles, and F. A. Nielsen. The real power of artificial markets. *Science*, 291:987–988, Feb. 2002.
4. L. J. Savage. Elicitation of personal probabilities and expectations. *Journal of the American Statistical Association*, 66:783–801, 1971.
5. E. Servan-Schreiber, J. Wolfers, D. M. Pennock, and B. Galebach. Prediction markets: Does money matter? *Electronic Markets*, 14, Sept. 2004.
6. A. Turing. Computing machinery and intelligence. *Mind*, 59:433, 1950.
7. L. von Ahn and L. Dabbish. Labeling images with a computer game. *Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 319–326, 2004.
8. L. von Ahn and L. Dabbish. Designing Games with a Purpose. *Communications of the ACM*, 51:58–67, August 2008.
9. J. Wolfers and E. Zitzewitz. Prediction Markets. *The Journal of Economic Perspectives*, 18(2):107–126, 2004.