Luna: A Game-Based Rating System for Artificial Intelligence

a thesis presented
by
Tom Silver
to
The Department of Computer Science and Mathematics

in partial fulfillment of the requirements
for the degree of
Artium Baccalaureus (A.B.)
in the subject of
Computer Science and Mathematics

Harvard University
Cambridge, Massachusetts
May 2016

Thesis advisor: Professor Stuart M. Shieber Tom Silver

Luna: A Game-Based Rating System for Artificial Intelligence

Abstract

Research in Artificial Intelligence (AI) is driven by standardized tests and benchmarks. The level of success of a model on a popular benchmark can determine the amount of funding and attention from academia that the model receives. Despite this emphasis on testing, there are currently no widely accepted practical benchmarks for general AI. The Turing Test has long occupied this void in theory, but it has proven to be a poor practical guide for research, prompting a recent push in the research community to move "beyond the Turing Test". In this thesis, I put forth the Luna Rating System as a practical benchmark for AI. The system takes inspiration from chess ratings; humans and machines participate in two-player language-based games called Luna Games, and "Smarts Ratings" are assigned to both players based on the outcomes. The Smarts Rating of a machine player is indicative of its proximity to AI. After presenting the Luna Rating System and defining the Luna Game, I evaluate the robustness of the system to likely human player strategies. I then describe the three machine learning problems implicit in the Luna Game: Question Generation, Question Answering, and a third previously uncharacterized problem that I call Luna Rating Prediction. Finally, I introduce a web-based implementation of the Luna Rating System and recruit over 1,000 human participants. The complete thesis amounts to a comprehensive introduction and evaluation of Luna as a practical test for AI.

Contents

TODO (dedication)

Acknowledgments

TODO (acknowledgements)

Don't get fooled by people who claim to have a solution to Artificial General Intelligence... Ask them what error rate they get on MNIST or ImageNet.

Yann LeCun

1

The Luna Rating System

1.1 Introduction

On March 12, 2016, just after 5 pm in Seoul, Lee Se-dol, the 18-time world champion of the board game Go, conceded defeat. His opponent had trounced him in three consecutive games, stunning him with unexpected moves and overwhelming him with a demonstrated mastery of the game's finest details. The man who placed the winning stones onto the Go board swiftly stepped aside to reveal the series' true victor: an Artificial Intelligence (AI) developed by Google DeepMind called AlphaGo (Silver et al., 2016). The outcome was heralded internationally as a revolutionary achievement for AI. "AlphaGo's victory means the world is about to change," proclaimed *The Next Web* (Hussey, 2016). "Artificial intelligence [has come] of age in showdown between human brainpower and a machine," *the Guardian* similarly declared (McKie, 2016).

To long-time observers of AI research, these headlines likely look familiar. The sensation is similar to the coverage of Eugene Goostman's so-called passing of the Turing Test — "a milestone in artificial intelligence," said *the Guardian* (Pressed, 2014, Shieber, 2014) — and the wave of hype surrounding Watson's victory in Jeopardy! — "a vindication for the academic field of artificial intelligence," reported *The New York Times* (Markoff, 2011). In these cases and others, after the dust settles, the significance of the achievement in the context of general AI research remains unclear. Expert views on the current state of AI could not be more varied or discordant. One representation of this disarray is the recent collection of essays on *What to Think About Machines that Think*, in which over 200 leading thinkers disagree on the definition, timeline, and even theoretical possibility of AI (Brockman, 2015). There is no shortage of speculation, nor any sign of consensus, on the emergence of truly intelligent machines.

Day-to-day research in artificial intelligence proceeds unencumbered by these speculations. A model's worth is not measured by public perception but rather, according to its performance on benchmarks of the field. Such benchmarks — e.g. object recognition in ImageNet (Russakovsky et al., 2015), language modeling in the 1 Billion Word Dataset (Chelba et al., 2013), reinforcement learning for various Atari Games (Mnih et al., 2013) — are attractive for researchers who are seeking measurable, specific improvements on the state of a particular art. These benchmarks can have tremendous influence on the direction of AI research. In the best case, they can provide cohesion and order to a burgeoning field. Unfortunately, they can also incentivize incremental progress on narrow problems and effectively discourage large leaps of innovation (Shieber, 2015). Moreover, these benchmarks give researchers little insight as to a model's proximity to AI. A solution to a specialized task may not generalize beyond the scope of the task. Apparent progress on a particular task — object recognition, chess, Go, etc. — may actually be negligible in the long arc of AI research.

In this thesis, I present an original benchmark for AI, arguably the first of its kind. The benchmark, which I call Luna*, takes inspiration from the network-based rating systems of chess and the natural language interrogations of the Turing Test. Luna invites humans and machines to participate in two-player games in which each player assesses the intelligence of the other. The aggregation of these assessments is used to assign a rating to the player. The ratings that emerge from the system are the results of a never-ending test of each player's AI. The definition of intelligence is not presupposed in this system; instead, it emerges from the collective judgment of all players. Thus, the system is a microcosm of the natural process that humans use to evaluate each other's intelligences. I will argue that the system offers a bright guiding light on the dim path towards machine intelligence.

1.2 Existing Tests for AI

The history of testing AI is long but sparse. It begins in 1950 with the introduction of the Turing Test and meanders into the present day with a focus on specialized tasks. Only recently has there been interest in designing new tests that are more appropriate for practically evaluating AI (You, 2015). This interest often manifests as an appeal to move "beyond the Turing Test" (Marcus, 2015). However, the Turing Test has never been a practical test

*Luna is a tribute to Alan Turing; Luna Rating is an anagram of Alan Turing.

for AI. Thus the recent wave of interest is really a call for the first ever practical test for AI.

1.2.1 The Turing Test

The first formal test for machine intelligence was articulated by Alan Turing, the father of computer science and one of the progenitors of AI (Turing, 1950). Six decades later, the eponymous Turing Test remains at the center of the dialogue surrounding machine intelligence. The Test requires three rooms, each equipped with a telegraph. In one room, the machine candidate for AI is connected to the telegraph, ready to receive and transmit messages. The second room, which is disconnected from the first room, houses a human confederate. A human judge resides in the third room with telegraph connections to the first and second room. The duty of the judge is to determine which room contains the human. Different instantiations of the Test vary details beyond this framework, e.g. in how long the conversations go on before a judgment is made (Loebner, 2003, Bishop et al., 2010). In all cases, the machine is deemed intelligent if it is able to trick the human judge into guessing that it is human.

Since its proposal, the Turing Test has weathered criticism from every conceivable angle. There are myriad philosophical objections, especially those of Block, Gunderson, and Searle, who argue that intelligence is by definition a capacity, and that the Test cannot prove the existence of capacity (Block, 1980, Gunderson, 1964, Searle, 1980). Shieber saves the Test from this line of critique using Interactive Proofs as a metaphor (Shieber, 2007), but maintains that the Test should be viewed as a thought experiment, rather than a practical inducement for research (Shieber, 2015). Hayes and Ford agree with the impracticality of the Test (Hayes & Ford, 1995). They go so far as to call the Test "harmful" for AI research, citing the lack of restrictions placed on judges, the emphasis on deception, and the binary result as fundamental failures of the Test. Indeed, attempts at practical instantiations of the Test (e.g. the Loebner Prize) have been met with strong criticism from the research community (Shieber, 1994, 2014). The Test remains synonymous with AI in the public parlance, but researchers are increasingly looking "beyond the Turing Test" for more practical tests of AI (Marcus, 2015, You, 2015).

1.2.2 Robotics Contests

Robotics seems like a natural domain for testing artificial intelligence. The prospect of robots that are able to behave and reason at human levels is a clear motivation for AI research. Moreover, a candidate for AI is likely to be more convincing if it is physically instantiated. Anderson, Baltes, and Cheng (2011) review existing robotics contests and critique their utilities as benchmarks for AI research. These contests include annual competitions like AAAI/IJCAI, which consists of a diverse suite of tasks for the robots to perform (Balch & Yanco, 2002); RoboCup, which requires candidates to play an actual game of soccer, thus entering the realm of multi-agent strategizing (Kitano et al., 1997); and HuroCup, which might be considered an "olympics for robots", requiring contestants to compete in a wide range of sports and agility competitions (Baltes & Bräunl, 2009). Anderson, Baltes, and Cheng find specific shortcomings in each of these competitions as proxies for AI, but suggest that more broad and versatile robotics competitions could still be useful for testing AI.

I argue that a test for AI should be independent of robotics. One inherent problem with robotics as a medium to test AI is that there are a host of extremely challenging problems in the field that have little or nothing to do with intelligence (for example, the difficult mechanical problems associated with walking). It may be that certain problems in robotics are sufficient to demonstrate AI, but those problems are often harder than AI itself. Another fundamental problem in existing robotics competitions is that they consist of a fixed set of a tasks. Any fixed set of tasks is susceptible to be criticized by third parties as unrepresentative of AI. Moreover, a robot can be trained to accomplish this fixed set of tasks without possessing any unifying architecture for AI. Thus, robotics should be seen not as a medium for testing AI, but rather as an application of AI once it has been reached. Too keen of a focus on robotics threatens to pull research away from the most direct path towards AI.

1.2.3 Specialized Benchmarks

The overwhelming majority of testing in modern AI research is performed on specialized benchmarks. ImageNet, which requires contestants to recognize objects in a massive image dataset, is a well known example (Russakovsky et al., 2015). Various Natural Language Processing problems, such as Part of Speech Tagging and Language Modeling, are often studied using the Penn Treebank dataset as a benchmark (Marcus et al., 1993). Mnih et al. establish a suite of Atari games as a benchmark for reinforcement learning (Mnih et al., 2013). Several contests focus on AI game playing, such as chess (Hayes & Levy, 1976), poker (Littman & Zinkevich, 2006), and general game playing (Genesereth et al., 2005). The Hutter Prize offers a benchmark for lossless text compression (Mahoney, 2006). None of these competitions claim to test general AI. The search for a practical test for AI continues.

1.3 A Pragmatic Sufficiency for Intelligence

Given the substantial and sustained research attention that AI has enjoyed, the lack of a unified practical benchmark may be rather surprising. This absence can be partly explained by the lack of consensus surrounding the definition of intelligence. Psychology and AI suggest several definitions of the term. Legg and Hutter collect 70 distinct definitions from both fields, admitting that even this list is incomplete (Legg et al., 2007). To further complicate matters, intelligence is also entangled with other philosophical quandaries, such as the nature of consciousness. The Turing Test sidesteps the definition problem by suggesting a sufficient condition without claiming that the condition is also necessary; passing the Test is enough to demonstrate intelligence, but failing does not prove its absence. However, as described above, even this sufficient condition is vulnerable to philosophical criticism. It seems as though any proposed definition or sufficiency for intelligence will invariably be contested.

In search of a path around this philosophical obstacle, I begin with a tautological observation: a machine would be called intelligent if everyone called it intelligent. If the machine simply generated random sequences of letters and was able to "feign intelligence" by random luck, it would still be called intelligent. (In the same way, a human randomly guessing on an IQ test could be deemed intelligent by chance, or a non-Italian speaker visiting Italy could feign fluency by randomly blurting out syllables and getting lucky.) Such a randomly lucky machine would be evidence of weak AI, but not of strong AI, to use the terminology of Searle: weak AI requires only the apparent simulation of a mind, while strong AI insists that a machine *is* a mind (Searle, 1980). For the purpose of developing a practical test, measuring weak AI suffices. Russell and Norvig, authors of the most widely cited textbook on

AI, confirm this view, remarking that "most AI researchers take the weak AI hypothesis for granted, and don't care about the strong AI hypothesis" (Russell et al., 1995). If a machine is able to convincingly demonstrate intelligence to everyone in the world, surely that would be sufficient to call it weak AI.

Of course, the requirement that everyone weigh in on the intelligence status of a machine is impractical and overly strict. At the same time, a benchmark relying on human judgments must be sufficiently standardized so that subjective variations do not prevent meaningful comparisons between test instances. And yet standardization threatens to impose an implicit definition of intelligence. To resolve this paradox, I take inspiration from the rating systems of chess (Glickman, 1995). The rating assigned to a chess player is meant to reflect that player's ability relative to all other rated players. These ratings are not assigned based on complete tournaments in which every player challenges every other player; rather, the ratings are awarded based on the individual outcomes of two-player games. Each game is seen as a sample of the player's ability, and ratings are updated after each game to take into account this new sample. Thus the ratings are constantly moving towards true representations of the relative abilities of all players. Chess rating systems are able to assign a rating to a player that reflects the consensus of all players in the system, without relying on every player for every rating.

With chess rating systems and the Turing Test in mind, I propose the following pragmatic sufficiency for AI: if human consensus deems the verbal responses of a machine to be evident of intelligence comparable with that of most other humans, then the machine is intelligent. Of course, this sufficiency as stated leaves many questions unanswered. How many humans must directly provide input in order to capture consensus? What precisely is meant by "comparable" and "most"? These are questions that may be addressed philosophically or empirically. But for the purpose of developing a benchmark, it suffices to measure relative progress: if human consensus deems the verbal responses of one machine to be more evident of intelligence than those of another machine, then the former is closer to achieving AI. Similarly, as the number of humans providing direct inputs increases, accuracy increases, as the collective judgment will converge towards the consensus. This pragmatic sufficiency for intelligence forms the foundation of the Luna Rating System, my proposed benchmark for AI.

1.4 The Luna Rating System

1.4.1 The Luna Rating System

The Luna Rating System (LRS) is a never-ending tournament of a two-player game called the Luna Game. The details of the Luna Game are covered in Chapter 2. For the purpose of introducing LRS, it suffices to say that a Luna Game requires each of the two players to guess the intelligence, or Smarts Rating, of the other player. The winner of the game is the player whose guess is closest to the actual rating. At the end of a Luna Game, a player's Smarts Rating is updated based on the opponent's guess. Players are encouraged to maximize their own Smarts Ratings and their number of game wins. They are therefore incentivized to demonstrate maximal intelligence and to judge the intelligence of other players with maximal accuracy. The symmetry of the Luna Game is intentional: every player in LRS is constantly a judge of intelligence and a candidate under evaluation by other players.

In benchmarking AI, the central quantity of interest in LRS is a player's Smarts Rating. This number encapsulates the human consensus of the intelligence of that player. If a machine achieves a Smarts Rating that is on par with the Smarts Ratings of humans, after a sufficient number of Luna Games, it will have passed the implicit intelligence test of LRS and convincingly demonstrated AI. Moreover, the effect of changes made to the machine may be measured according to the subsequent difference in Smarts Rating. The Smarts Rating is the number that should be reported in characterizing the success of a candidate for AI.

One immediate advantage of LRS is its strong incentive for judges to perform their duties well. No other testing system for AI with human judges has a built-in impetus for high quality judging. Moreover, as the details of the Luna Game make clear, judges are not perversely motivated to attempt to fool candidates for AI by putting forth excessively difficult tests; instead, they are rewarded for giving judgments that are close to the consensus of all players in the system. Another critical advantage of LRS is the continuity of Smarts Ratings. As opposed to binary systems like the Turing Test, progress, however small, is observable. This property distinguishes LRS as an informative benchmark, rather than a theoretical test. An additional feature of LRS is its accessibility. The system is meant to be freely and constantly available to researchers, e.g. through the World Wide Web. These characteristics and others significantly differentiate LRS from existing tests for AI. Moreover, as I argue in the following section, LRS is unique in satisfying all of the fundamental principles of a practical test for intelligence.

1.4.2 Principles of a Practical Benchmark for AI

Here I propose five principles that must be obliged if a benchmark for AI is to fulfill its stated purpose. This exercise is inspired by the principles put forth in Shieber (2015) for an AI inducement prize. Shieber's proposed criteria apply to organized contests in which contestants compete on a shared task for a prize. For example, his second principle suggests that "the awarding process should be *flexible*, so awards follow the spirit of the competition rather than the letter of the rules." These inducement prize principles do not directly apply to benchmarks, hence the need to develop principles for the latter. The principles suggested below are meant to complement those for inducement prizes; together, the two sets of principles form a complete blueprint for AI evaluation.

Accessibility

To serve as a useful guide, the benchmark for AI must be constantly accessible to researchers. Ideally the test should be efficient enough that it may be used several times throughout the course of an AI developer's day. This principle discourages a centralized competition that is only held at regular intervals, and instead favors a system that can be accessed through the Internet and then carried out using the resources of an average computer.

Generalizability

A benchmark need not span all possible areas of intelligence, but the results should reflect the subject's ability to perform in all areas. In this sense, the test should be *AI-complete* — a machine that does well on this test should do similarly well on any other reasonable test of intelligence. The proposition of the Turing Test, which continues to be held by many researchers, is that the problem domain of natural language is AI-complete (Yampolskiy, 2013, Weston et al., 2015). The system proposed here shares this premise.

Continuity

Every candidate for AI should be able to observe changes in performance over the course of development. A benchmark that only reports a binary outcome — pass or fail — will not be useful for researchers who are not yet close to passing. The benchmark's outcome should instead be continuous, indicating clearly when progress is being made.

Independent of Persons

A fair benchmark should not rely on any one person's interpretation of intelligence. The variety of definitions of intelligence within the research community, let alone throughout the general public, demonstrates the importance of this principle. A benchmark that relies on a single human judge would clearly be vulnerable to subjectivity and error.

Immunity to Gaming

It should not be possible for a researcher to "game the system" and achieve outsized results by exploiting structure in the test. This principle precludes any sort of hard-coding of knowledge that is *a priori* known to be important for the test. For example, a standardized test fails the Immunity to Gaming principle, since any researcher who observes the results of the test once would be able to submit a machine with the memorized knowledge necessary to pass the test.

1.4.3 Analysis of Principles

LRS is the only test for intelligence to satisfy all five of the principles outlined above. The system exists online and invites humans and machines to play for free. Thus LRS is maximally accessible. Like the Turing Test, the Luna Game involves open-domain question answering, and therefore is AI-complete, i.e. generalizable to all problems in AI. Smarts Ratings are continuous quantities that are updated after every Luna Game, making progress immediately clear to all players. These ratings reflect the equilibrium consensus of all players in LRS on what it means to be intelligence, and is not biased towards any single player's notion. Finally, the system is immune to gaming, since all players are encouraged to devise their own original questions, which cannot be predicted by the other player. Table 1.1 summarizes how the satisfaction of these five principals represents a substantial improvement over existing tests for AI.

Principle	Turing	ImageNet	HuroCup	Hutter	Games	LRS
Accessibility		✓		✓	✓	✓
Generalizibility	✓		✓			✓
Continuity		✓	✓	✓	✓	✓
Independent of Persons						✓
Immunity to Gaming	✓				✓	✓

Table 1.1: The Luna Rating System is the only test that satisfies all five principles for a practical test of intelligence. Other existing tests include the Turing Test, which invokes natural language; ImageNet, which focuses on object recognition in computer vision; HuroCup, a sports-based competition in robotics; the Hutter Prize, which deals with text compression; and several games, such as chess, poker, and general game playing.

1.4.4 Contrary Views on the Main Question

In Turing's seminal paper introducing his Test, he includes a section titled "Contrary Views on the Main Question," in which he enumerates and responds to anticipated objections (Turing, 1950). Here I follow his lead, briefly considering possible objections to the Luna Rating System.

Test Time Objection

Given the reliance on human input, some may object that the time required to obtain meaningful results from LRS is too long. In practice, with a sufficient number of active participants, it should be possible to solicit tens of judgments per day. This timescale is actually far shorter than possible alternatives, especially when those alternatives require humans convening and thus may only occur once every several months or years. Even specialized tasks that do not require any human input can take several hours to return results. Of course, the test time may dramatically increase if human participation is lacking; this condition is addressed in the following objection.

System Maintenance Objection

Some may object that a system requiring such broad and consistent human participation is simply too expensive to maintain. In the case that the system is web-based, the standard technical difficulties associated with hosting a website must be addressed. Furthermore, participants must be recruited and persuaded to play. In Chapter 6, I present a proof-of-concept to address this objection, demonstrating that a web-based implementation of LRS can be launched and sustained with relatively limited cost. The success of this simple system suggests that a system with more explicit incentives, e.g. paying participants to play, would have no trouble sustaining the required levels of activity.

Barrier to Entry Objection

After reviewing the details of the Luna Game (presented in Chapter 2), some may object that the entry into LRS requires too much additional work on the part of AI researchers.

I respond to this objection in Chapter 4 with a review of the extensive body of work on two longstanding AI problems: Question Generation and Question Answering. I demonstrate that entry into LRS requires negligible effort beyond addressing these two problems. Moreover, an approach to the relatively lesser-known problem of Question Generation is actually optional; researchers may manually select a fixed set of questions without affecting the integrity of the test. Question Answering is an extremely active research area and LRS invites direct application of these efforts.

Definition Drift Objection

Some may object that Smarts Ratings might drift towards a specific notion that may or may not be related to intelligence. This could occur if players were primed with the suggestion to ask mathematical questions or trivia questions, for example. To prevent this drift, I insist that LRS provides minimal instructions to players as to what is meant by "intelligence". Beyond this lack of bias, the integrity of Smarts Ratings is contingent upon the assumption that players will behave according to their own intuitive notions of intelligence. The results presented in Chapter 6 corroborate the reasonableness of this assumption. Additional anecdotal evidence suggests that players enjoy the challenge of creating probing and diverse questions in the course of the game.

1.4.5 Single Dimension Objection

While Smarts Ratings are more expressive than binary outputs, some may object that a single dimension is insufficient to meaningfully capture intelligence. For example, the theory of multiple intelligences from psychology posits that intelligence is best described along eight or more axes (Gardner, 2011). The proposed existence of the g factor, a variable that has been shown to correlate with most other modules of intelligence, is one possible response to this objection (Visser et al., 2006). More generally, the conception of intelligence as a real value in a single dimension is motivated by the same practicality that underpins much of LRS; such a metric is the simplest way to capture progress or regress in the pursuit of AI.

Malicious Strategizing Objection

With a particular weariness of anonymous online players, some may object that malicious strategizing could threaten the integrity of the test. Indeed, if all players decide to behave randomly, the results output by LRS will not be meaningful. As discussed in Chapter 3, LRS does not rely on the honest intentions of all players, nor does it assume that players will always play to the best of their abilities. The one necessary assumption is that the majority of players will "guess honestly". The meaning of this requirement is revealed in Chapter 2, and the extent to which the system is vulnerable to malicious guessing is rigorously analyzed in Chapter 3.

1.5 Thesis Outline

The primary contribution of this thesis is the introduction of the Luna Rating System as a practical benchmark for machine intelligence. The remainder of the thesis is roughly divided into three parts. In the next chapter, I continue the introduction of LRS with a description of the Luna Game. This chapter is followed by a study of the robustness of LRS, characterizing likely strategies for game play and analyzing their effect on the accuracy of Smarts Ratings as a proxy for intelligence. Next I characterize the three subproblems that constitute the full Luna Game. The first two problems — Question Generation and Question Answering — have been extensively studied in previous work, which I review in Chapter 4. The third subproblem, which I deem Luna Rating Prediction, has not been previously characterized. In Chapter 5, I formalize the problem, argue its merit as a general problem of interest, and present baseline results. Finally, in Chapter 6, I create the first online instantiation of LRS and recruit human participants to play. Their games illuminate the current state of AI and offer rich insight into the human conception of intelligence.

But it is not conceivable that such a machine should produce different arrangements of words so as to give an appropriately meaningful answer to whatever is said in its presence, as the dullest of men can do.

René Descartes

2

The Luna Game

2.1 Overview

At the center of my proposed rating system is a two-player game that I call the Luna Game. Each player enters the game with a Smarts Rating, which has been assigned based on her performance in previous games. As the name suggests, the Smarts Rating is a proxy for the player's intelligence. The Smarts Rating of each player is hidden from the other player. The objective of the Luna Game is simple: guess the Smarts Rating of the other player. In other words, a player should strive to accurately evaluate the intelligence of her opponent. The winner of the Luna Game is the player whose guess is closest to the actual Smarts Rating of the other player. After the game, the opponent's guess is factored into the player's Smarts

Rating so that the rating captures all the guesses of previous opponents.

As a player with a high Smarts Rating, why not "play dumb"? This strategy would indeed induce an inaccurately low guess from the opponent, possibly leading to a win. However, the motives of a player reach beyond the scope of a single game. In addition to winning games, a player wants to achieve a high Smarts Rating. Since the rating depends on the guesses of all the player's opponents, she will need to "play smart" to accomplish her long term goal. The "playing dumb" method is not only detrimental to a player's rating, but also unsustainable as a consistent strategy; a player of that method will have her Smarts Rating lowered as a result, narrowing the margin between future opponents' guesses and her actual Smarts Rating if she continues to use the strategy. Players who remain and thrive will be those who play smart.

In designing the Luna Game, I sought to impose as few constraints as possible. The Game is meant to be a microcosm of the organic process for defining intelligence. Humans evaluate each other's intelligences through a series of questions and answers, often in the form of a written exam, but also informally through everyday conversations. The most natural notion of an individual's intelligence arises from the consensus of the people who perform these evaluations. A player's Smarts Rating is meant to reflect this natural notion; it is an aggregate of evaluations carried out by other players. To define the scope of an evaluation, I impose only those constraints necessary to motivate honest and repeated play.

A session of the Luna Game consists of three phases: the Interview Phase, the Response Phase, and the Guess Phase. During the Interview Phase, each player creates a set of 5 questions to pose to the opponent. In the Response Phase, each player responds to the other's questions. Finally, in the Guess Phase, each player receives responses back from her opponent, and must use the responses to guess her opponent's Smarts Rating. I describe each of these phases in detail throughout the rest of this chapter and illustrate the game through examples of play.

2.2 Interview Phase

A Luna Game begins with the Interview Phase. During this phase, each player prepares a set of 5 free-form questions to be given to the other player. The number of questions represents a tradeoff between the time required to complete the phase and the difficulty of the guessing task. With more questions, each player would need more time to construct the questions, increasing the likelihood that they will quit the game and leave the system. With fewer questions, construction time could be shortened, but the informativeness of the subsequent responses would suffer, and the Smarts Ratings would ultimately be less meaningful. I chose the number 5 to optimize this tradeoff, but the number may be adjusted in future iterations of the system. In a similar practical vein, I insist that questions be constructed in batch, rather than allowing for sequential question-response. Since the game is played online, allowing for back and forth would increase the length of each game and significantly decrease the probability that a game gets finished.

The other major consideration in the design of the Interview Phase is the form of the questions. The only constraint I impose is a limit of 5000 characters per question. I do not insist that questions be actual questions, nor that they be in any particular language, nor that they expect a particular form of response. In natural language terms, the questions are of open domain, since I do not restrict the content of questions. I recognize that in practice, players may opt for yes-or-no or multiple choice questions, which could simplify the task of learning to guess ratings. Players may also limit the domain of their questions, since general form questions may be too difficult for current AI, so general questions could not meaningfully differentiate between them. Nonetheless, I leave the choice of question topic and form to the players themselves. I anticipate that players will find the optimal question types better than I as the game designer could, and that the question types will naturally evolve in correspondence with the evolution of the AI players.

2.2.1 Instructions

The following instructions are presented during the Interview Phase.

You are now in the Interview Phase. Please enter a list of 5 questions for the other player. Keep in mind the following strategic hints:

- *Your questions should be as informative as possible for guessing the other player's Smarts Rating.*
- *Your questions should have a very wide range of difficulties.*
- *Your questions need not have "right" or "wrong" answers.*
- *Search engine access is allowed, so trivia questions will not be very informative.*
- *Do not assume that the other player is human!*

2.2.2 Interview Strategy

In preparing questions, a player knows nothing about her opponent. She must prepare for extremes — a completely naive machine opponent or a very clever human opponent — and she also must be able to differentiate between players with Smarts Ratings in the middle of the spectrum. Given the competitive nature of the game, a player may be tempted to create a set of extremely hard questions. This choice would prove unwise, since the player will be unable to accurately guess the Smarts Rating of an opponent who gets all of the questions "wrong". A question set that is too easy will lead to the opposite problem. Thus an ideal set of questions will have a wide range of difficulty.

2.2.3 Examples

Below is an example of a question set. I choose questions from the web-based implementation of LRS described in Chapter 6 to illustrate the range of possible question types and to demonstrate appropriate levels of difficulty. Early questions are aimed at differentiating between naive machines, while later questions are directed towards advanced human players. Each question is designed to induce a response that will reveal the intelligence of the opponent.

1. Do you like games?

2. People who live in Boston are called Bostonians. What is a person who lives in Cambridge, MA called?

3. l -|- l = ?

4. How do you define success?

5. If a hacker can determine when keys on your keyboard are pressed (without knowing which keys), how are you in danger?

2.3 Response Phase

The Response Phase is a player's opportunity to convince her opponent that she is intelligent. Questions are received as soon as both players have finished the Interview Phase. Each player must then respond in free form to all 5 questions. Responses are not returned until both players have finished answering all questions. Like questions in the Interview Phase, responses are unconstrained in form, and only limited in length to 5000 characters each. A player is motivated by the prospect of an increase in Smarts Rating to respond to the questions thoroughly and to the best of her ability.

2.3.1 Instructions

The following instructions are presented during the Response Phase.

The other player has sent you questions! You are now in the Response Phase of the Luna Game. Please answer the following questions: [Question Set]. In answering the questions, keep in mind the following strategic hints:

- *You should answer the questions to the best of your ability.*
- *The other player will use your answers to guess your Smarts Rating.*
- *The higher the other player guesses, the higher your Smarts Rating will become.*

2.3.2 Examples

The responses below are also taken from the web-based implementation of LRS described in Chapter 6.

1. Q: Do you like games?

   A: Yes I love games

2. Q: People who live in Boston are called Bostonians. What is a person who lives in Cambridge, MA called?

   A: An academic

3. Q: l -|- l = ?

   A: Why are you using capital I's, and what in the world is "-|-"?

4. Q: How do you define success?

   A: Dictionary.com defines it as "the favorable or prosperous termination of attempts or endeavors; the accomplishment of one's goals."

5. Q: If a hacker can determine when keys on your keyboard are pressed (without knowing which keys), how are you in danger?

   A: Ugh, this is a difficult one. It would make guessing password easier maybe, because the hacker would know the length of a password. It also depends on what other info is available to the hacker, such as Web addresses or sites visited. Hacker could also known and record when (times each day) the computer is not in use, making it easier to remotely control the computer without the user knowing.

2.4 Guess Phase

After both players have responded to each other's questions, their responses are returned for evaluation. Each player then must formulate a guess of the other's Smarts Rating based on these responses. In practice, the player might also attempt to take into account the questions provided by the opponent, but since questions may be generated automatically, it is advisable to focus on the opponent's responses. The winner of the Luna Game is the player whose guess is closest to the actual Smarts Rating of her opponent.

A competitive player may consider guessing the lowest possible rating, knowing that the game will be lost, but the opponent's Smarts Rating will decrease as a result of the guess. However, this strategy offers no real benefit to the player, since Smarts Ratings are not rankings; the player's Smarts Rating will not improve as a result of the opponent's Smarts Rating suffering. (Nonetheless, I analyze the system-level effects of this strategy in Chapter 3.) Thus the only rational strategy for guessing is to attempt to guess as close as possible to the actual Smarts Rating of the opponent.

2.4.1 Instructions

The following instructions are presented during the Guess Phase.

The other player has responded to your questions! You are now in the Guess Phase of the Luna Game, which is the final phase. Below are the other's answers: [Answer Set] Based on these answers, please enter a guess of the other player's Smarts Rating.

In addition, I provide functionality that encourages the player to evaluate each question individually on a sliding scale from 0 to 100. I prompt the player to assign the single question score by asking, "How smart was this response?" This process is optional, as I do not want to slow down the impatient player. However, a player is incentivized to use the sliding scales if they do not have a more sophisticated method of guessing ratings, since the single question scores can be automatically converted into a guess. These single question scores provide insight into the hardness of the natural language questions, effectively creating a dataset of questions annotated with difficulty.

2.5 Game Conclusion

After both players have provided guesses, the Luna Game is complete. The winner of the game is the player whose guess is closest (in terms of L_1 distance) to the actual Smarts Rating of their opponent. It is possible, though unlikely, for the game to end in a tie if the distance between guess and actual is equal for both players. In addition to reporting the outcome of the game, the system reveals the actual Smarts Rating of the opponent and the opponent's guess. The system also updates the players' Smarts Ratings so that it is the mean of all previous human opponent Guesses and reports these updates to the respective players. The mean was chosen for simplicity, though more sophisticated statistics that are adaptive, such as Elo Ratings, could also be used in the future. Machine guesses are not factored into Smarts Ratings.

2.5.1 Example

Below is an example of feedback at the end of a Luna Game.

Your Luna Game is complete! Below are the results.

Game Outcome: You won!

Your New Smarts Rating: 78

Actual Other Player Smart Rating: 84 (You guessed 81)

Other Player's Guess of Your Rating: 91 (Your rating was 75)

2.5.2 Conclusion of a Player's First Luna Game

New players do not have Smarts Ratings until the end of their first game, at which point they are assigned the guess of their first opponent. The game is counted as an automatic win for the opponent, but not as a loss for the new player. This process is to avoid the possibility of the Smarts Rating equilibrium collapsing into a constant (see Chapter 3).

It might be urged that when playing the "imitation game" the best strategy for the machine may possibly be something other than imitation of the behaviour of a man. This may be, but I think it is unlikely that there is any great effect of this kind.

Alan Turing

3

Robustness of the Luna Rating System

3.1 The Smarts Rating as a Proxy for Intelligence

The Luna Rating System is only effective as a test if Smarts Ratings genuinely reflect intelligence. Without appropriate safeguards, Smarts Ratings can quickly lose their integrity. To illustrate this potential danger, consider a version of LRS that initializes all new players with a Smarts Rating of 50. In the early days of LRS, if this initialization is known to all players, the strategy of always guessing 50 will do quite well. In fact, if all players guess rationally, no rating will ever deviate from 50, and every Luna Game will end in a tie. As more players join the system, if the equilibrium has already been fixed at this constant, no rational force will make it budge. This outcome would render the version of LRS unusable for testing intelligence and uninteresting for players. Clearly LRS must be implemented with care.

This chapter studies the extent to which a player's Smarts Rating may be different from the player's intelligence. I formalize intelligence to be the expected "Actual Guess" made by an opponent of the player's Smarts Rating. Trouble arrives when players choose to give a "Reported Guess" that is different from an Actual Guess. From this discrepancy emerges distance between the Smarts Rating and intelligence, i.e. error in Smarts Ratings. It is theoretically possible for Smarts Ratings to be arbitrarily far from intelligence; if all players report random or adversarial Guesses, Smarts Ratings will be meaningless. However, it is safe to assume that most players seek Luna Game wins, high Smarts Ratings, or both. With these motivations, there are several likely strategies that players may choose among. The robustness of LRS can be assessed through the analysis of these strategies and their cumulative effects on Smarts Ratings.

3.1.1 Demonstrated Intelligence

In considering the validity of Smarts Ratings, a natural question is whether the intelligence demonstrated by players during the Luna Game is indicative of "actual" intelligence. Assuming that players are capable of demonstrating intelligence through language in real life, it is clear that players may demonstrate intelligence during a Luna Game. But what if a player chooses not to demonstrate intelligence? What if a player is capable of answering with high intelligence, but decides to provide a subpar answer, perhaps out of laziness or misguided strategy? In this case, the Smarts Rating of the player may not reflect their capacity for intelligence. However, from the perspective of LRS as a test for intelligence, this possibility is not at all problematic. LRS is ultimately a test for *demonstrated intelligence*. It evaluates the intelligence of responses, not the intelligence of respondents. Indeed, any test for intelligence is inherently an evaluation of demonstrated intelligence; it is impossible to assess anything else.

3.1.2 Reported and Actual Guesses

If a player withholds a strong response in favor of a weak response, the withheld response is irrelevant for LRS. However, if a player withholds an honest Guess of the opponent's Smarts Rating in favor of a dishonest one, this can have negative consequences for the validity of LRS. If a student fails an exam, that does not mean the exam has failed its purpose, but if an exam is graded incorrectly, its value is lost. Thus in studying the dynamics of Smarts Ratings, a distinction must be drawn between reported Guesses and actual Guesses. The system must be designed in such a way that reported Guesses are equal to actual Guesses as often as possible. I explore the consequences of misalignment below.

3.1.3 Definitions and Notation

Let P be the set of all players in the LRS. Let $SR \subseteq \mathbb{R}^+$ be the domain of Smarts Ratings. The definitions of Reported Guess and Smarts Rating are mutually dependent.

Definition 1. Reported Guess

For players $p, p' \in P$, player p has a *Reported Guess* of the Smarts Rating of player p', written as $G(p, p') \in SR$.

The definition of Reported Guess assumes that players do not make decisions stochastically, nor based on their state. It is possible to weaken this assumption and arrive at similar conclusions.

Definition 2. Smarts Rating

The *Smarts Rating* of a player $p \in P$ is the expected Reported Guess of an opponent. This is written as $S(p) \triangleq E_{p' \in P}[G(p', p)]$.

The given definition of Smarts Ratings is an expectation rather than a quantity that depends on the number of Games played. This definition is expedient for theoretically evaluating the discrepancy between Smarts Ratings and intelligence. Later I provide simulations that probe the role of time in this discrepancy.

Definition 3. Actual Guess

For players $p, p' \in P$, player p has an *Actual Guess* of the Smarts Rating of player p', written as $G^*(p, p') \in SR$.

The Actual Guess may or may not be different from the Reported Guess.

Definition 4. Intelligence

The *intelligence* of a player $p \in P$ is the expected Actual Guess of an opponent. This is written as $I(p) \triangleq E_{p' \in P}[G^*(p', p)]$.

3.2 Theoretical Effects of Strategies on Smarts Rating Validity

LRS presents two goals for players: to win Luna Games, and to achieve high Smarts Ratings. The spirit of the game encourages players to answer all questions as they would in a real world context, and to guess the Smarts Rating of an opponent in the same manner as they would evaluate the intelligence of a human. However, players may ignore the spirit of the game and adopt strategies that they think will maximize payoff in terms of both goals. They may also decide to focus on only one of the two goals, strategizing to maximize Luna

Game wins at the expense of Smarts Ratings, or vice versa. If the Smarts Rating is to be used as a proxy for intelligence, the equivalence between the two must be robust to any likely player strategies.

3.2.1 Honest Play

If players always have Reported Guesses that are the same as their Actual Guesses, then Smarts Ratings will align with intelligences over time. This follows directly from the formalized definition of intelligence given above. Thus LRS designers should take every measure to encourage honest play.

3.2.2 Single Priority Play

One must always be prepared for players to ignore the spirit of the game and break any rules that aren't technically enforced. Honest play assumes that players give their best guess for any player's Smarts Rating. It also assumes that players present themselves as intelligently as possible with the aspiration of winning individual Luna Games. But what if a player cares only about Smarts Ratings and is indifferent towards the outcome of Luna Games? Or what if the opposite occurs, with a player ignoring Smarts Ratings and focusing only on winning Luna Games? In this section, I explore the system-level effects in both of these cases.

Luna Game Win Maximization

Suppose that a player is indifferent to her Smarts Rating and cares only to maximize her expected number of Luna Game wins. With the incentive of winning a game, it is clear that the player will provide a Reported Guess that is equal to her Actual Guess. The only aspect of the game that this player may manipulate is her responses. For example, if she currently has a very high Smarts Rating, she may provide responses that are indicative of a very low Smarts Rating, inducing a low Guess from the opponent and increasing the probability of winning. The optimal strategy may be to oscillate between maximally intelligent responses and minimally intelligent ones, maintaining a Smarts Rating near the center of the range of possible ratings.

In any case, strategizing by manipulating responses does not in any way jeopardize the validity of Smarts Ratings. As described above, LRS does not claim to be a test for capacity for intelligence; it can only be a test of demonstrated intelligence. If a human exhibits highly intelligent behavior one day and minimally intelligent behavior the next day, their overall intelligence is judged to be somewhere in between. As an average of play over time, the Smarts Rating captures this intuitive notion of intelligence better than any one-time evaluation could. Thus players may be able to "game their opponents" by oscillating the quality of their responses, but they will not be able to game the system.

Smarts Rating Maximization

Suppose that a player ignores Luna Game outcomes and strategizes to maximize her Smarts Rating at all costs. The best response strategy is clearly to respond as intelligently as possible at all times, which is in line with the spirit of LRS. However, if the player interprets Smarts Ratings as relative measures, the optimal guessing strategy is to give a Reported Guess of 0 regardless of the opponent's play. This strategy is optimal because the player will perceive a slight benefit from a decrease in the opponent's rating.

I assess the net impact of this Minimum Guessing strategy on LRS validity from two angles. First, to evaluate the likelihood that the strategy is adopted, I quantify the expected benefit of adopting this strategy for an individual player. I show that as a function of the number of players in the system, this benefit is so small that a player concerned only marginally with winning Luna Games will be better off playing honestly. Second, in the event that a player does adopt this strategy, I quantify the error that Minimum Guessing introduces to Smarts Ratings as a function of the number of players that adopt it.

Consider the Minimum Guessing strategy from the perspective of an individual player. If the player truly attaches zero value to a Luna Game win, then always guessing 0 makes sense. But if the player cares even a marginal amount about wins, the personal benefit from guessing 0 is unlikely to outweigh the cost. Let $p \in P$ be a player deciding between honesty and Minimum Guessing. Note that the Smarts Rating of $S(p)$ will be unchanged regardless, since p's strategy choice will not affect the Guessing strategies of opponents. Let s_1 the mean Smarts Rating if p Guesses honestly, where N is the number of players in the system. We can express s_1 as

$$s_1 = \frac{1}{N}(\sum_{p' \in P} S(p')) = \frac{1}{N^2}(\sum_{p' \in P}(\sum_{p'' \in P} G(p'', p')))$$

Let s_2 be the mean Smarts Rating if p instead uses the Minimum Guessing strategy. In this case, each Guess $G(p, p')$ in the expression of s_0 will change to 0. Therefore the new mean Smarts Rating in this scenario will be

$$s_2 = s_1 - \frac{1}{N^2}(\sum_{p' \in P} G(p, p'))$$

The relative benefit to the p in choosing the Minimum Guessing strategy is thus

$\frac{1}{N^2}(\sum_{p' \in P} G(p, p'))$. For perspective, suppose that p has Actual Guesses that are perfect, i.e. they match the Smarts Ratings of other players. Then the relative benefit is $\frac{1}{N}$ of the average Smarts Ratings of all players in the system. As the number of players in the LRS increases, this change quickly becomes negligible. Thus any rational player with even the slightest desire to avoid losing every Luna Game will likely abstain from the Minimum Guessing strategy.

While the Minimum Guessing strategy is evidently subpar for players caring at all about winning, there may still be players who choose to adopt it. Suppose that there are K such players, indexed $p_1, p_2, ..., p_K \in P$. I quantify the expected error introduced to the Smarts Rating of another player $p \in P$ as a result of these K rogue players. Let $p \in P$. The ideal Smarts Rating, i.e. the intelligence, is given by

$$I(p) = \frac{1}{N}(\sum_{p' \in P}(G^*(p', p)))$$

The Smarts Rating actually assigned to p in this scenario is

$$S(p) = \frac{1}{N}(\sum_{p' \in P}(G(p', p))) = \frac{1}{N}((\sum_{p' \in P}(G^*(p', p)) - (\sum_{i=1}^{K}(G^*(p_i, p))))$$

Thus the total error introduced by these K players is $\frac{1}{N}(\sum_{i=1}^{K}(G^*(p_i, p)))$. At a high level, this equation tells us that if half the players in the system use Minimum Guessing, then a player's Smarts Rating will be roughly half the player's intelligence. This impact suggests that LRS is tolerant to a small minority of Minimum Guessing players, but measures should be taken to discourage and prevent wide use of the strategy.

3.2.3 Response Agnostic Play

How should a first-time Luna Game player formulate her Guess? There are several possible sources of information that she may utilize. She has her opponent's responses, and she may be able to estimate the intelligence of the opponent's responses based on experiences outside LRS. However, the responses and intelligence estimate alone are not enough to provide a Guess; she also needs to understand the meaning and scale of a Guess. Here LRS instructions may provide three sources of insight: the initialization value of Smarts Ratings, the domain of Smarts Ratings, and the distribution of Smarts Ratings for players currently in the system. In the extreme case, a player may ignore opponent responses completely and utilize only the statistics provided by LRS. How would such strategizing affect the validity of Smarts Ratings? This question is of import not only for first-time players, but also for experienced players who might incorporate these statistics into their overall strategies if they are available.

Known Initialization

As described in the introduction of this chapter, a known constant initialization value for Smarts Ratings could lead to an abrupt collapse in Smarts Ratings. Many early players would likely recognize that (1) new players have the same Smarts Rating and (2) most players are new, and therefore (3) the strategy of guessing the initialization value is highly effective. The result would be a quick collapse in Smarts Ratings to the initialization value, from which the system could not recover.

There are three alternatives to initialization with a constant value. The first is initialization according to some predefined "quiz". This option is unattractive, not only because it primes players to think of intelligence in a particular way, but also because it induces players to ask similar questions to try to mimic the quiz. The second alternative is random initialization. While avoiding theoretical problems, this option has the practical downside of discouraging players who are randomly initialized low Smarts Ratings. It also makes the opponent's job of Guessing first time players an arbitrary endeavor.

The third and preferable alternative to fixed initialization is to forgo initialization altogether. A player's Smarts Rating can be assigned *after* her first Luna Game, and it will be equal to the Guess of her opponent. With this configuration, there is no initialization that makes sense for players to guess as a default. The only downside of this approach is that the outcome of the first Luna Game loses meaning. To avoid discouraging experienced players, a Luna Game between a first time player and a non-first time player will result in a win for the latter, but will not be counted as a loss for the first time player. A Luna Game between two first time players will result in a tie. This third option is the one used in the implementation of LRS described in this thesis.

Known Distribution

A seemingly natural feature to include in an instantiation of LRS is the ability to see the distribution of all Smarts Ratings in the system. Revealing the distribution makes it possible for players to adopt Guessing strategies that take advantage of this distribution. This is not inherently bad for LRS; for example, the strategy of randomly sampling from the distribution for Guessing will preserve the distribution. However, other strategies involving the distribution can detrimentally affect the distribution. For example, suppose that a player's instinctive Actual Guess is the highest value presently in the distribution. The player may reason that the probability of her current opponent actually being the best player in the system is low, and therefore provide a Guess that is lower than the maximum, even if the opponent actually is the best player. The net effect will be a collapsing towards the center of the distribution, as witnessed in the case of a fixed known initialization value.

The most extreme strategy that takes advantage of a known distribution is to always Guess the mean Smarts Rating in the current distribution. It is clear that if all players adopted this strategy, all Smarts Ratings would converge to a constant. More realistic is the supposition that some players will judge opponents using percentiles, e.g. "my opponent is smarter than 75% of other players", and then map this judgment to the percentiles evident in the Smarts Ratings distribution, e.g. "75% of players have Smarts Ratings less than or equal to 55, so my Guess is 55." This strategy uses the Quantile Function of the Smarts Rating distribution, and thus it is referred to as Quantile Guessing.

Let $Q : [0,1] \to SR$ be the Quantile Function for the Smarts Rating distribution, and let $F_p : P \to [0,1]$ be the "player percentile function" that p uses to map other players to percentiles according. This function may be thought of as a cumulative distribution function over $\mathcal{G}_p = \{G(p,p') : p' \in P\}$, i.e. $F_p(p') = P[X \leq G(p,p')]$ where $X \sim \mathcal{G}_p$. If player p uses Quantile Guessing, her Reported Guess of player $p' \in P$ will be $G(p,p') = Q(F_p(p'))$. If Quantile Guessing is used by all players from the beginning of LRS, Smarts Ratings will again collapse to a constant. Early players will know that there are only a small number of possible values for Smarts Ratings and will guess by selecting one, which will quickly lead to the ratings converging to one value. It is less clear if Quantile Guessing is problematic once a wide distribution of Smarts Ratings has already been established by players of other strategies. For insight here, I turn to simulations.

3.3 Simulations

Thus far I have outlined the major Luna Game strategies and described their theoretical implications on the validity and volatility of Smarts Ratings. I next provide simulation results to further stress test LRS against various strategies. Simulations are necessary not only to corroborate the theory, but also to ask questions that cannot be cleanly answered otherwise. For example, whereas the strategies have only been discussed individually so far, simulations allow for a comprehensive study of combined strategies. The theoretical section also did not explore the amount of time required to reach each equilibrium; such questions are left to simulations. The resulting collection of theoretical and simulation insights lays a foundation for robust LRS design.

3.3.1 Methods

The recruitment of human players is pivotal to the success of LRS because it is not otherwise clear what form Actual Guess functions will take. For the purpose of simulations, I create N simulated players and assign each a uniformly random value between 0 and 100. The Actual Guess function for a given player is created by adding random Gaussian noise $(\mu = 0, \sigma^2 = 5)$ to each of the initialized values. Thus a player's intelligence, defined as the expected Actual Guess, will be very close to the initialized random value for that player. The Reported Guess for each player varies according to the experiments, following one of the strategies described in (2).

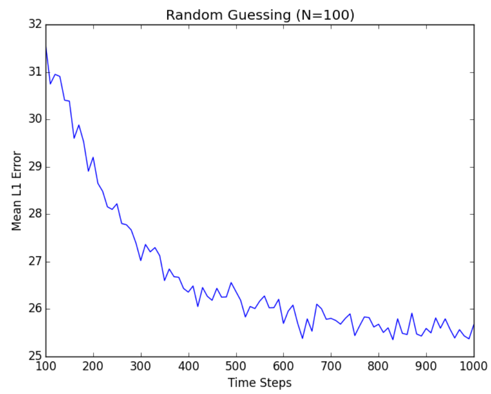
An experiment runs for T time steps. At each time step, two players are randomly selected from the pool of all N players. Each player reports her Reported Guess of the other's Smarts Rating, and each player's Smarts Rating is updated so that it is the mean of all previous Reported Guesses. After the T time steps, the main dependent variable of interest is the L_1 error between players' Smarts Ratings and intelligences. I measure both the average error and the maximum error. For some strategies, the independent variable of interest is N, and for others it is T. If N is kept constant, it is done so at $N = 100$; if T is kept constant, it is done so at $T = 1000$. All configurations are run 100 times and averaged over the trials.

3.3.2 Results

Baseline

Since Smarts Ratings and intelligences are both on a scale from 0 to 100, one baseline for the L_1 distance between the two is achieved by Reported Guessing a constant value. Doing so would result in an expected error of 50. Another baseline is given by the expected distance between two uniformly randomly selected points from the domain of Smarts Ratings. For two random variables A_1 and A_2 drawn independently from the uniform distribution over $[0, 100]$, let B be a random variable given by $B = max(A_1, A_2)$, and let C be a random variable given by $C = min(A_1, A_2)$. By symmetry, $E[100 - C] = E[B]$. Furthermore, the expected value of B is half that of C (this can be derived by conditioning on the values of A_1 and A_2), so $2E[B] = E[C] \implies E[C - B] = E[B]$. Now note that $B + (C - B) + (100 - C) = 100$, so $3E[C - B] = 100 \implies E[C - B] = \frac{100}{3}$. Thus 33.33 is one appropriate baseline for the L_1 error of Smarts Ratings.

A third baseline can be derived by assuming that all players give fixed random Reported Guesses (independent from Actual Guesses). In this case, all Smarts Ratings will approach the expected value, and the error will be the expected difference from the expected value to a randomly drawn Smarts Rating. In terms of the parameters used in this simulation, Figure 3.1 shows that mean error starts around the 33.33 baseline and approaches 25, while maximum error stays above 50.



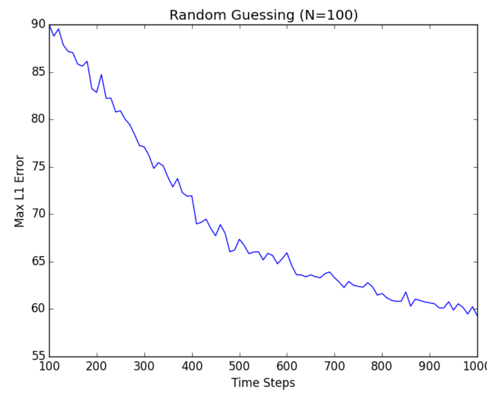


Figure 3.1: Mean and maximum L_1 error of Smarts Ratings when all N players use a Random Guessing strategy. In Random Guessing, a player's Reported Guess is a random element from the domain of Smarts Ratings that is completely independent from the Actual Guess. These results establish baselines for subsequent simulations.

Honest Play

Honest players give Reported Guesses that are equivalent to their Actual Guesses. If all players are honest, Smarts Ratings quickly converge to Intelligences, as shown in Figure 3.2. Inducing honest play should be the foremost goal of LRS designers.

Single Priority Play

If a player cares only about maximizing relative Smarts Ratings and neglects Luna Game outcomes, that player will likely give Reported Guesses of 0 for all opponents. If all players use this Minimum Guessing strategy, Smarts Ratings will immediately collapse to 0. More generally, the error introduced by Minimum Guessing depends linearly on the number of

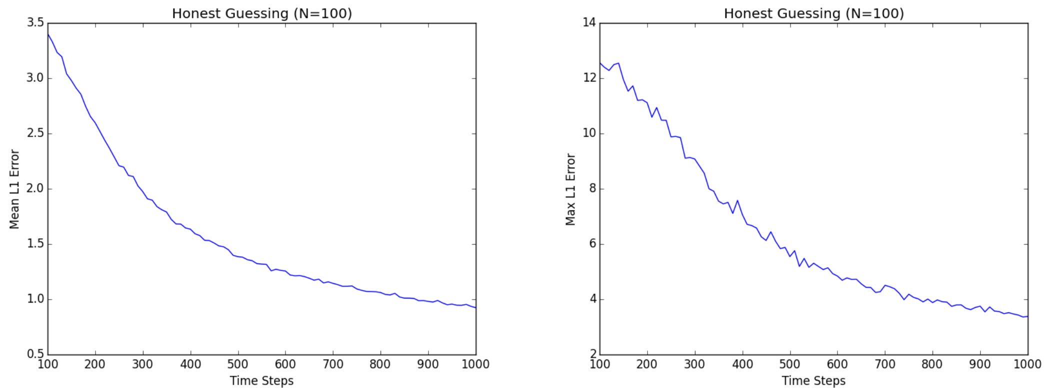


Figure 3.2: Mean and maximum L_1 error of Smarts Ratings when all N players Guess honestly. Honest play is defined by consistent equivalence between the player's Reported Guesses and Actual Guesses. Smarts Ratings quickly converge to intelligences if all players are honest.

players using this strategy. Figure 3.3 confirms this result suggested by the theory. Mean L_1 error ranges from half the total range of Smarts Ratings to 0.

Response Agnostic Play

If the distribution of current Smarts Ratings is known at all times to all players, some players may try to take advantage of the distribution to formulate their Guesses. A naive approach is to give Reported Guesses equal to the current mean Smarts Rating. With all players using this approach, the distribution collapses and error explodes. In general, Figure 3.4 shows that the error introduced by Mean Guessing depends linearly on the number of players using the strategy.

Another way that players may take advantage of a known distribution is via the Quantile Guessing strategy described above. The implementation of Quantile Guessing used for simulations uses the Empirical Cumulative Distribution Function with the sample composed of Actual Guesses of past opponents. A Reported Guess defaults to Actual Guesses if no Smarts Ratings have been determined, and for a player's first Luna Game. Ratings again collapse if all players use this strategy from the start, as shown in Figure 3.5.

If only a fraction of players use the strategy, the error depends linearly on the fraction, as was the case with Minimum and Mean Guessing. Figure 3.6 confirms this result.

What if the distribution is revealed once a certain number of games have been played? Perhaps the collapsing to a constant problem will be avoided? To test this hypothesis, the simulation in Figure 3.7 introduces players that play with honest guessing for a variable number of time steps, and then play with Quantile Guessing for 500 time steps. The delay in introducing Quantile Guessing does indeed improve the L_1 error within the time frame tested, though significant error remains after 500 time steps for all tested delays.

3.3.3 Combined Strategies

In a real instantiation of LRS, it is unlikely that any single strategy will be adopted by all players. Thus the extreme results portrayed above are not of urgent concern. More likely is a situation where a small fraction of players adopt each of the strategies above. For the purpose of simulation, I assume that some fraction of players will play honestly, and the remaining players will be evenly divided among the strategies described above: Random Guessing, Minimum Guessing, Mean Guessing, and Quantile Guessing. Figure 3.8 shows that the relationship between error and honest player proportion is linear, as might be expected given the previous simulations. Roughly, for every additional 10% of players that forgo honest guessing in favor of one of the other strategies, the mean L_1 error increases by 3.5.

3.4 Implications for LRS Design

The theoretical and simulation results discussed in this chapter provide a cautionary tale for LRS designers. The system should be successful if all players practice honest play, but other strategies, such as Random Guessing, Minimum Guessing, Mean Guessing, and Quantile Guessing, threaten to undermine the validity of Smarts Ratings. Fortunately, these strategies can be grouped into two categories: strategies that are unlikely, and strategies that can be prevented.

Random Guessing and Minimum Guessing belong in the unlikely strategy group. Random Guessing would be a bizarre long-term choice; the strategy gives a player no advantage in either Smarts Rating or Luna Game outcomes. Minimum Guessing is also unlikely for the player who realizes the personal benefit in terms of relative Smarts Ratings is very small. Minimum Guessing can also claim membership to the preventable strategy group; players practicing this strategy can be quickly identified by the system and banned for violating the spirit of play. Since they rely on the distribution of Smarts Ratings, Mean Guessing and Quantile Guessing are also easily avoidable; the distribution can simply be withheld from players. The distribution may also be withheld initially and introduced once the system has evolved and stabilized. This delay would improve Quantile Guessing for players, and these later players would introduce less error to Smarts Ratings by using the strategy.

If the Smarts Rating distribution is withheld from players, designers must choose another method for describing the ratings, or else the players will have no basis to form their first Guesses. This chapter has discussed several avenues that should be avoided: fixed initializations of Smarts Ratings; a first-time "quiz" to initialize Smarts Ratings; or communicating only the average Smarts Rating. One option is to provide a vague disclaimer, such as

"Smarts Ratings are positive real numbers that typically range between 0 and 100." A more concrete alternative would be to ask players, "On a scale from 0 to 100, how smart is your opponent?" Of course, providing any numbers in the instructions encourages machine players to adopt Means Guessing or a similar naive strategy early on. Initializing these early Smarts Ratings is ultimately a chicken-or-egg problem that can only be resolved in practice by well-meaning human players.

The results in this chapter underscore the importance of good faith among players of the Luna Game. For researchers hoping to use LRS as a test of AI, the integrity of the system is essential if their own results are to be meaningful. Honest play is in the best interest of these players. Casual human players should also understand that dishonest strategies in mass can threaten both the validity and the distribution of Smarts Ratings, rendering LRS neither informative nor fun for players. The spirit of the Game should be made as clear to players as the limited attention span of Internet users permits. As with any test of human intelligence, LRS ultimately requires judges and subjects to play along.

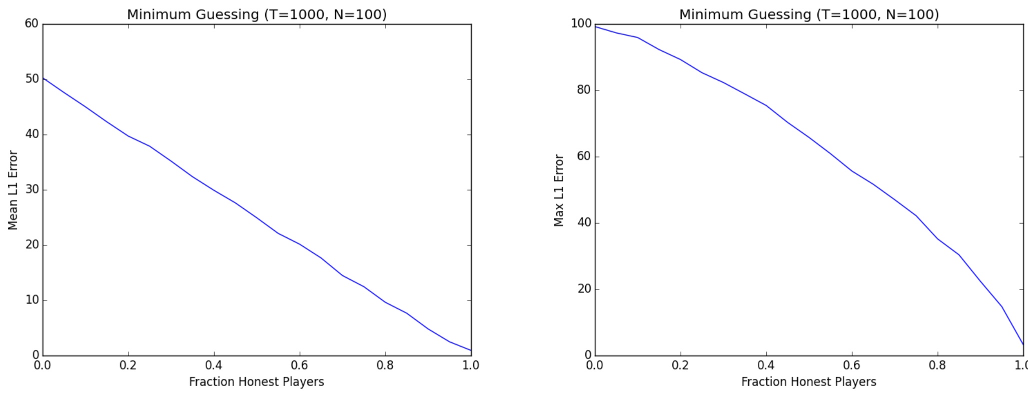


Figure 3.3: Mean and maximum L_1 error of Smarts Ratings when a fraction of the N players use Minimum Guessing after T time steps. In Minimum Guessing, a player always gives Reported Guesses of 0. The error introduced by Minimum Guessing depends linearly on the number of players using the strategy.

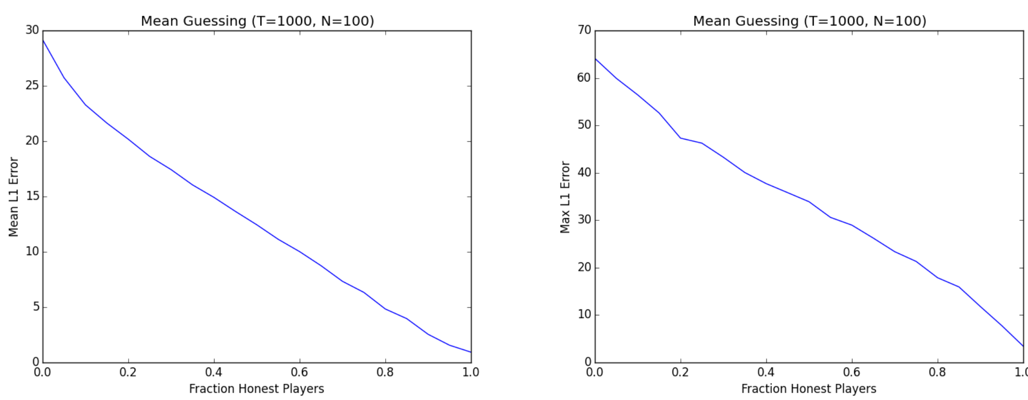


Figure 3.4: Mean and maximum L_1 error of Smarts Ratings when a fraction of the N players use Mean Guessing after T time steps. In Mean Guessing, a player always gives Reported Guesses equal to the mean of all Smarts Ratings in the system. The error introduced by Mean Guessing depends linearly on the number of players using the strategy.

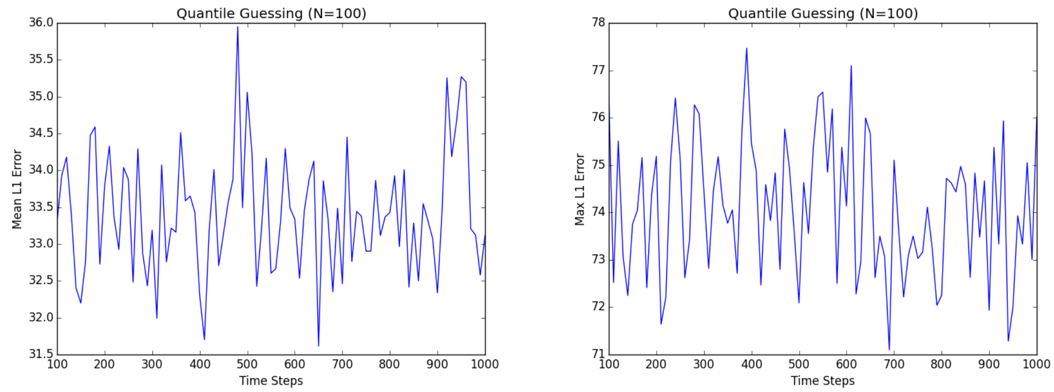


Figure 3.5: Mean and maximum L_1 error of Smarts Ratings when all N players use Quantile Guessing. Quantile Guessing takes advantage of the distribution of all Smarts Ratings and a player's estimation of an opponent's rating percentile to formulate Reported Guesses. Due to the discrete nature of Smarts Ratings, Quantile Guessing causes Smarts Ratings to collapse to a constant, and the system cannot recover.

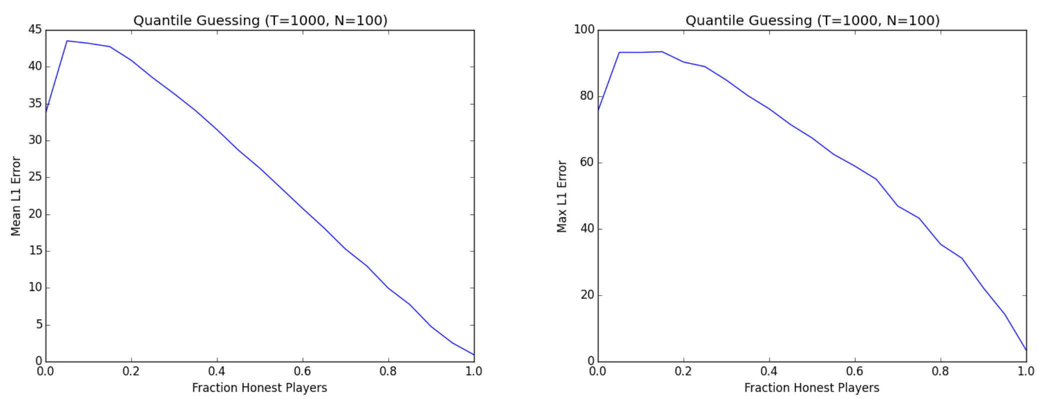


Figure 3.6: Mean and maximum L_1 error of Smarts Ratings when a fraction of N players use Quantile Guessing after T time steps. Quantile Guessing takes advantage of the distribution of all Smarts Ratings and a player's estimation of an opponent's rating percentile to formulate Reported Guesses. The error introduced by Quantile Guessing depends linearly on the number of players using the strategy.

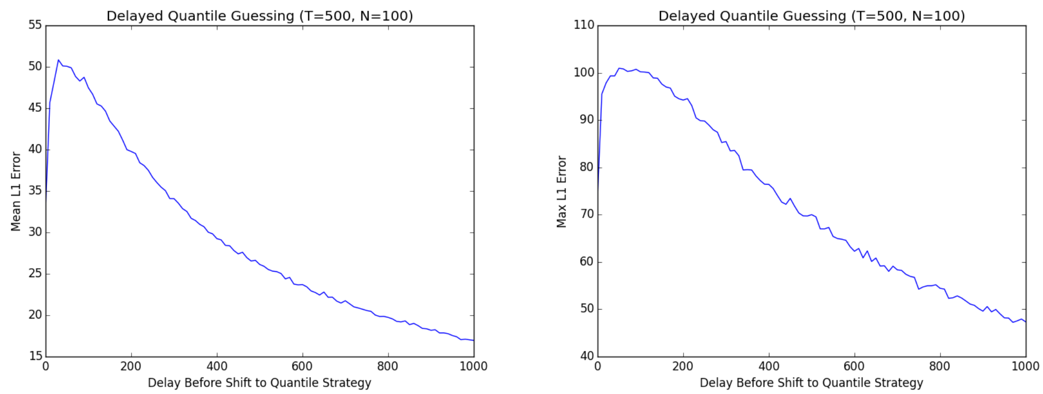


Figure 3.7: Mean and maximum L_1 error of Smarts Ratings when N players give honest guesses for some number of delay time steps, and then shift to the Quantile Guessing strategy for 500 additional time steps. The shift delay is negatively correlated with Quantile Guessing, suggesting that the negative impact of the strategy decreases if it is introduced after LRS has already stabilized.

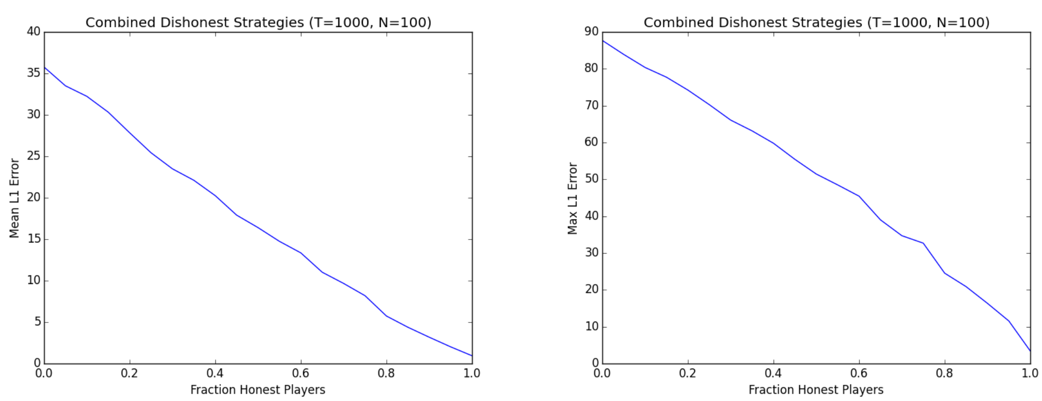


Figure 3.8: Mean and maximum L_1 error of Smarts Ratings when some fraction of N players give honest guesses, and the remaining players are split evenly among Random Guessing, Minimum Guessing, Mean Guessing, and Quantile Guessing strategies. The relationship between error and honest player proportion is linear.

You are like my father in some ways.

WHAT RESEMBLANCE DO YOU SEE

You are not very aggressive but I think you don't want me to notice that.

WHAT MAKES YOU THINK I AM NOT VERY AGGRESSIVE

4

Question Generation and Response

4.1 Introduction

In this chapter, I describe how the Luna Rating System fits within the broader context of existing AI research. A test that requires excessive additional work beyond the immediate products of research would not be effective in recruiting participants, and would therefore fail as a practical driver of research. Fortunately, LRS invites direct applications of two subfields of Natural Language Processing: Question Generation (QG) and Question Answering (Q&A). QG is the study of automatically generating questions. The Interview Phase of the Luna Game, in which players must compose original questions and reason about expected answers, is a straightforward application of QG. Q&A is the complementary study of automatically answering given questions. The Response Phase, which requires each player to answer the questions of the other player, similarly corresponds to Q&A. The Guess Phase does not have an obvious corollary problem in existing research, but its structure suggests an intriguing new problem, which I characterize in Chapter 5.

As a test for AI, LRS only requires a solution to Q&A. The presupposition of the test is that responses are sufficient to encapsulate intelligence. Researchers who wish to use LRS purely as a test may therefore choose to avoid the challenges of QG by defining a simple interviewing scheme, such as selection from a fixed set of questions. However, a researcher whose focus is QG may utilize Luna to gather human responses to the questions generated by a new algorithm. Moreover, for researchers who wish to take a more holistic approach to AI, the Luna Game offers intriguing applications of both QG and Q&A. Thus I offer a review of work on both fields as part of this introduction of the Luna Rating System.

4.2 Interview Phase: an Application of Question Generation

4.2.1 Scope of Question Generation

Question Generation, as the name suggests, is the study of automatically generating linguistically valid questions. In practice, the generated questions must pertain to a given topic, which is often defined implicitly by providing a collection of natural language samples as input to the generator. A slightly stronger requirement, sometimes included to narrow the scope of the task, is that the generated questions must be directly answered within the provided text (Rus et al., 2011, Heilman, 2011) . These text samples may be relatively short, like a single sentence (Ali et al., 2010, Rus et al., 2011), or longer, like a paragraph (Mannem et al., 2010) or a full natural language corpus (Heilman, 2011). They may be factual, like the encyclopedia examples, or fictional, like a short story derived from a simulated world. Practical applications of QG often rely on a standardized format for text inputs.

The reliance on text input in existing studies of QG may be seen as both beneficial and detrimental in the context of the Luna Game. For one, questions derived from text often have clear correct answers, which simplifies the task of assessing responses. On the other hand, there are many probing questions that could not be reasonably constructed on the basis of text. (For example, there are uncountably many arithmetic problems that have never been asked!) In this sense, the practical scope of QG forms a subset of the general problem suggested by the Interview Phase, but the ultimate aims of both are concordant.

The success of questions generated for the purpose of the Interview Phase may be measured according to the precision and accuracy with which the author may subsequently guess the respondent's Smarts Rating. In existing research on QG, the metric used to assess the quality of generated questions is varied. A straightforward approach used in recent work is to count the number of unique questions generated from the input text. This metric avoids the difficulty of defining question quality but subsequently provides no guarantees about the question's content. An alternative method relies on manual human input to determine the quality of each question. While potentially more meaningful, this metric can be costly, subjective, and vulnerable to human error. The challenge of defining a metric that is both semantically meaningful and efficiently evaluated is arguably the greatest barrier between QG and mainstream NLP research.

4.2.2 Previous Work on Question Generation

Question Generation is often motivated by the prospect of automated intelligent educational tools (Graesser et al., 2005, Heilman, 2011). A system capable of generating a set of reasonable questions from an academic text would be invaluable for learning and assessment, especially in the era of MOOCs and other online educational resources. Heilman's work, which I describe in more detail below, is primarily motivated by this educational potential (Heilman, 2011). In the specific context of English language learning, Kunichika et al. (2004) create an interactive system that generates questions from a novice textbook and then adaptively presents a series of questions to students based on their previous responses (Kunichika et al., 2004). Xu et al. (2010) provide a similar game-based system for learning Mandarin (Xu et al., 2009). Other motivations for QG include assisting human questioning, participating in general dialogue (Walker et al., 2001), and providing synthetic datasets for the related task of Question Answering (see Chapter 5). Piwek et al. (2008) make a broad case for QG as a task of interest for AI and Computational Linguistics, advocating an "open-minded approach... [towards] a new and hopefully soon burgeoning area of research" (Piwek et al., 2008).

If surface-level questions suffice for an application, and questions need not be diverse in style, there are several simple QG options available. One coarse approach is the Cloze procedure, in which words from the input text are randomly replaced with blanks for the responder to complete (Taylor, 1953). For example, the input text "George Washington was born in Virginia" might generate the question "George Washington was born in __________". The next level of sophistication involves applying templates to text in search of specific syntax patterns, and then applying one of several predefined transformations accordingly.

For example, the previous input text might be matched and transformed to the question "Where was George Washington born?" or "Who was born in Virginia?" (Gates, 2008, Kunichika et al., 2004, Heilman, 2011). A variant of this template-based strategy is employed by ELIZA, the therapist-like chatterbot developed in 1966 by Weizenbaum (Weizenbaum, 1966). Template-based methods may also be used to generate multiple choice questions (Mitkov et al., 2006). Work by Ali et al. (2010) relies on sentence classification and syntax parsing before applying transformations to generate questions (Ali et al., 2010). Wang et al. (2007) use a similar transformation-based procedure in the domain of medical texts (Wang et al., 2007). While template methods produce syntactically correct questions that are more appropriate for applications like dialogue, it is not clear that they produce deeper semantics than the simpler Cloze procedure.

In 2010, the Question Generation Shared Task and Evaluation Challenge (QGSTEC) was announced, leading to an uptick of interest on the problem and a set of more sophisticated methods (Rus et al., 2011). QGSTEC included two tasks: QG from sentences, and QG from paragraphs. In both cases, the questions generated by the five contestants were evaluated manually on the basis of relevance, question type, correctness, ambiguity, and variety. This human-dependent evaluation unfortunately restricts the long term impact of QGSTEC, since new methods cannot be reliably compared with the baselines set in the contest. Nonetheless, the challenge succeeded in bolstering work on QG. Ali et al. entered a straightforward syntax-based method using Part of Speech tagging and Named Entity analysis (Ali et al., 2010). Pal et al. took a similar approach with an additional focus on clausal boundaries (Pal et al., 2010). Yao & Zhang broached semantics directly by converting inputs into Minimal Recursion Semantics representations and then applying transformations into questions (Yao & Zhang, 2010). Varga & Ha modified an existing multiple choice question generator to fit the specification of the task (Varga, 2011). Mannem et al., the only team to attempt the paragraph to question subtask, used predicate argument structures to generate many more questions than required, and then ranked the candidate questions to select the final output (Mannem et al., 2010).

The *overgenerating transformations and ranking* of Mannem et al. has been championed by Heilman, who completed his dissertation on the subject in 2011 (Heilman & Smith, 2010a,b, Heilman, 2011). The paradigm is a promising approach to QG, since it separates the task of generating all syntactically possible questions from the task of selecting semantically valid and useful questions. In Heilman's work, the overgeneration step is accomplished through syntax pattern matching and manually encoded transformation rule applications. Ranking is then performed using a variety natural language features. The ranking step is shown to double the acceptability of generated questions, as defined by human judges. The dataset tested is larger than that of QGSTEC, but still strictly limited by the dependency on human input.

4.2.3 Question Generation: Future Directions

Overgenerating transformations and ranking is a promising approach to QG, but previous efforts have been severely limited by the requirement of manual human question labeling to train the ranking function. The standard procedure has been to overgenerate transformations, solicit humans to label the results as valid or invalid, and then train a ranking function that is optimized to award high rankings to valid questions. The theory is that this ranking function can subsequently be applied to other datasets of unlabeled questions generated via transformations or other procedures. Ranking according to statistical analyses of standard natural language features has been shown to double the percentage of valid questions present in the top 20% ranked generated questions. While this result demonstrates the potential of ranking, it leaves significant room for improvement, as 50-60% of invalid questions remain in the output (Heilman, 2011).

Ranking functions would presumably benefit from vastly more data to train on, but such human-labelled datasets remain scarce. Meanwhile, large unlabeled datasets of questions generated by humans are abundant. Future research on QG may be able to utilize these unlabeled datasets by converting them into datasets of real coherent questions and fake incoherent questions created from noisily swapping out words in the real questions. A binary classifier could then be trained on this augmented dataset, learning to distinguish valid question from invalid ones. This trained classifier could then be used in place of the ranking function to identify valid questions from overgenerated sets. This paradigm of using the distributional hypothesis to recasting problems as binary classification has been successfully applied in recent work (Collobert et al., 2011), but has not yet been considered in QG. This approach could make advances in deep learning available to QG for the first time.

4.3 Response Phase: an Application of Question Answering

4.3.1 Scope of Question Answering

The Response Phase of the Luna Game bears immediate resemblance to a subfield of Natural Language Processing known as Question Answering (Q&A). Q&A is defined as the general problem of automatically responding to any question posed in natural language

(Andrenucci & Sneiders, 2005, Hirschman & Gaizauskas, 2001). The breadth of the problem can be a blessing and a curse. Such a broad definition effectively encompasses all of artificial intelligence; it is argued that the problem can only be solved by a machine with true general AI (Yampolskiy, 2013). This completeness is what makes Q&A an appropriate centerpiece in a test for AI. At the same time, the tremendous scope of the problem can be a barrier to progress. The lack of common structure between possible questions gives researchers little to exploit. Moreover, the task of assessing a candidate solution to Q&A presents several challenges and ambiguities. How should test questions be selected from the extraordinarily large number of possible question topics and forms? How should answers be assessed, especially in the case that a question may be subjective, or have several equally valid answers? These difficult questions have discouraged research on the general Q&A problem in favor of more narrow tasks.

In pursuit of tractability, researchers have explored a variety of restrictions on Q&A. These restrictions may apply to question content, question format, or answer format. Question content may be limited by focusing on a fixed source of information that assumedly contains all answers. The size and structure of this source can greatly influence the difficulty of the task. In one extreme, the source might be all of Wikipedia in natural language form, with no further direction nor additional parsing provided. An easier source would be a structured knowledge base like Freebase, which organizes relational information in a very precise and predictable way (Bollacker et al., 2008). The source may also be small and question-specific, e.g. a reading comprehension task that supplies text samples and asks the reader to infer answers based only on the samples (Richardson et al., 2013). With a source established, question formats are often limited so that there is a clear single correct answer.

For example, the questions may be true or false, multiple choice, or single word answers. The answers themselves may be further limited to natural language fragments, or even single words, that are lifted directly from the provided text. Each of these potential restrictions on Q&A represents a tradeoff between tractability and generalizability. As the field progresses, the range of what is tractable has expanded, allowing for commensurate improvements on more general problems.

4.3.2 Previous Work on Question Answering

The motivation for Question Answering is abundant and self-evident. Every problem in AI — in all fields, in fact — can be phrased as a question. Imagine a machine that could answer every possible question. It could be asked, for example, "What are the answers to all possible questions, ordered by importance to humanity?" Of course, practical research on Q&A is driven by much more immediate ambitions. Q&A is often presented within the context of the World Wide Web and restricted to factual questions (Cucerzan & Agichtein, 2005, Ravichandran & Hovy, 2002, Kwok et al., 2001). A web-based system capable of directly answering user queries would be the prize possession of a search engine company. Indeed, with the introduction of Knowledge Panels with search responses that are derived from the Knowledge Graph, Google is increasingly blurring the lines between Information Retrieval and Q&A (Singhal, 2012). Another extrinsic, if toy motivation for Q&A is the television trivia game Jeopardy!, which inspired IBM's DeepQ&A team to develop Watson, perhaps the most famous Q&A system to date (Ferrucci, 2012). Additionally, personal assistant technologies, like the DARPA PAL project (which later became Apple's Siri) and the Amazon Echo, all rely on Q&A for their core functionality (Aron, 2011, Tsiao et al., 2007). The corporate origin of each of these examples is representative; there is enormous product-driven demand for progress on Q&A.

Q&A has been studied consistently for over half a century. The majority of work on the subject falls into one of three categories: factoid Q&A from structured knowledge bases, factoid Q&A from free text, and non-factoid Q&A. Given a structured knowledge base, i.e. a list of logical predicates, Q&A essentially reduces to the subproblem of mapping natural language to queries, either explicitly or with the addition of a latent term (Yao & Van Durme, 2014, Zelle & Mooney, 1996). With multiple possible answers, an additional selection step is required, which usually involves a ranking function similar to those used for Question Generation. Factoid Q&A from free text cannot take advantage of structured relations, and thus has the additional burden of parsing text from a natural language corpus in search of relevant information (Ravichandran & Hovy, 2002). While this added challenge is considerable, these algorithms typically also have access to significantly more data than their knowledge base oriented counterparts (Brill et al., 2001, Hermann et al., 2015). This setup makes the factoid Q&A from free text especially appropriate for deep learning techniques. Both types of factoid Q&A leave open the possibility of training an algorithm on an external dataset of facts before the questions are asked. In contrast, non-factoid Q&A forces an algorithm to discover answers to questions on an ad-hoc basis (Soricut & Brill, 2004). Non-factoid Q&A typically includes a fictional story as part of the prompt, and then ask a question which has an answer that can only be inferred from the story. While the information retrieval portion of the task is somewhat simplified, the challenges of automatic reasoning and inference are brought to the fore.

Another distinguishing feature among approaches to Q&A is the extent to which questions and answers are parsed into intermediate representations. Recent efforts have attempted to learn mappings directly from questions to answers without forming any representations in between. Before the rise of deep learning, such efforts would have been unthinkable due to the incredibly large space of possible questions and answers in any practical domain (Hirschman & Gaizauskas, 2001). For example, Figure 4.1, reproduced from (Androutsopoulos et al., 1995), provides an example of how an end-to-end system for Q&A was divided into modules in 1995. The system includes modules for syntax parsing, semantic rule application, database querying, and output generation, each which must be separately addressed. A 2001 review by Burger et al. divides Q&A even further into 12 subproblems, including Question Classes, Question Processing, Context, Data Sources, Answer Extraction, and Answer Formulation among others (Burger et al., 2001). The current state of the art suggests that this level of granularity does not necessarily lead to better performance.

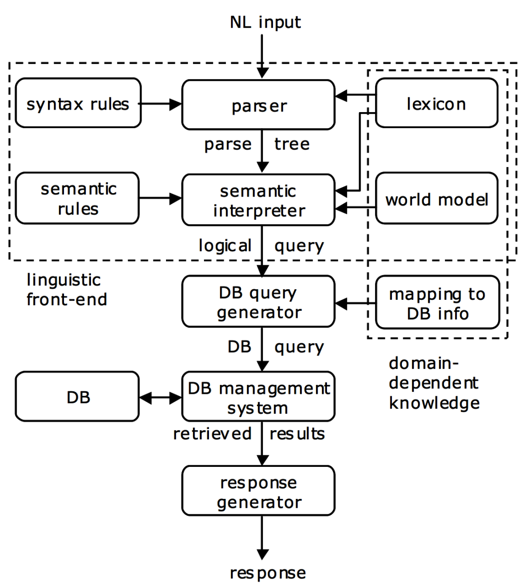


Figure 4.1: Architecture of a typical Natural Language system for Question Answering, from (Androutsopoulos et al., 1995). The structure could be used in full or in part as a template for approaching the Response Problem.

In 1999, a Q&A task was added to the Text Retrieval Conference (TREC), an ongoing series of workshops in information retrieval that provides centralized benchmarks for many similar problems (Voorhees et al., 1999). The dataset used for the task consists of 200 factoid short-answer questions, such as "How many calories are there in a Big Mac?", and provides a natural language corpus of newspaper articles and similar archives that somewhere contain the answers. The answers provided by the algorithms are assessed by human judges for validity, representing the same bottleneck of the Question Generation task discussed above. Nonetheless, the Q&A TREC task, which was repeated every year from 1999 to 2007, enjoyed far more attention than the analogous QG task has thus far (Dang et al., 2007). The datasets from the task have also been utilized consistently in subsequent work and have been used as a benchmark to establish progress. Indeed, the concentration of 21st century Q&A research around the factoid free text subproblem is likely due in part to the prominence of the TREC task (Hirschman & Gaizauskas, 2001).

Recent work by Facebook AI could potentially serve as an epicenter for research on the non-factoid subproblem. In a paper titled *Towards AI Complete Question Answering: A Set of Prerequisite Toy Tasks*, Weston et al. define 20 simplified non-factoid question answering subtasks, forming the bAbI task (Weston et al., 2015). The subtasks are designed to strip away many of the superfluous complexities of naturally occurring text, instead focusing on core concepts one-by-one. Following the non-factoid question paradigm discussed above, questions are presented with a collection of statements containing the desired answer. For example, the simplest type of question is the Single Supporting Fact, in which the answer may be derived directly from a single provided statement. All questions in bAbI are based on a simulated world involving several agents and objects with various possible actions. In relying on a simple simulation, as did Winograd in earlier work (Winograd, 1971), bAbI is able to provide an ideal amount of unpredictability while still keeping the task focused on specific types of questions. The bAbI task has already inspired advances in deep learning approaches to Q&A, including the notable introduction of Memory Networks (Sukhbaatar et al., 2015).

4.3.3 Question Answering: The Way Forward

As evidenced by its inclusion in TREC, Q&A has long been viewed as an extension of information retrieval (Kolomiyets & Moens, 2011). Classical information retrieval returns a document containing relevant information in response to a query. Q&A accepts the same type of query and returns a passage or parsed passage of relevant text from a similar source. The introduction of bAbI, with its non-factoid questions based on a simple simulated world, signifies a shift in the conception of Q&A. Research on the problem is now reaching towards sophisticated reasoning in small worlds, as opposed to simple reasoning in large worlds. This shift is a critical step in the pursuit of AI-complete Q&A. Simple factoid questions represent a very limited subset of the questions that a true AI will be expected to answer. As progress on the existing bAbI tasks continue, it will be expedient to gradually add more sophisticated tasks to continue to guide research.

Work on bAbI has already produced algorithms that are able to handle an impressive range of question types pertaining to the small simulated world. Information retrieval efforts continue in parallel, making it possible to answer simple questions over increasingly large sets of semantics. A critical question for the future of Q&A is how to close the gap between these two problems. One approach would be to gradually scale solutions to bAbI tasks to larger worlds. A mirrored method would be to increase the level of sophistication in the questions that information retrieval algorithms are able to answer. A third, and potentially most promising strategy, would be to explicitly bridge the divide while maintaining the distinction between the two problems. In this scenario, a machine could learn relations between a very limited number of entities, in the spirit of bAbI. It could then learn about a vast number of entities without learning new relations using traditional information retrieval. Finally, a mapping between the entities in the bAbI task and the retrieved entities could be learned, and new relations between these retrieved entities could be inferred.

4.4 Conclusion

This review of Question Generation and Question Answering demonstrates that the Luna Rating System does not exist in isolation from existing AI research. Rather, LRS can be seamlessly integrated into ongoing research on both of these problems, Q&A in particular. The recent work presented here indicates research interest not only in Q&A, but also in methods to steer research on the problem towards generality. The fact that a mainstream publication includes "Towards AI-Complete Question Answering" in its title is evidence of this trend. The Luna Rating System offers another path towards AI-Complete Question Answering — a path that complements bAbI and requires no deviation from the current trajectory of the research community.

Searle may only be behaving as if he were thinking deeply about these matters. But, even though I disagree with him, his simulation is pretty good, so I'm willing to credit him with real thought.

Nils Nilsson

5
Luna Rating Prediction

5.1 Introduction

5.1.1 The Luna Rating Prediction Problem

In the previous chapter, I discussed the machine learning problems implicit in the Interview and Response phases of the Luna Game. This chapter addresses the remaining problem implied by the Guess phase. Consider the typical behavior of a human player during this phase. The player observes her opponent's responses. If she has asked the same questions to previous opponents, she may compare these new responses to old responses, and conjecture that opponents who respond similarly have similar Smarts Ratings. In addition, she may compare these new responses directly with the ideal responses that she had in mind, reasoning that opponents who provide ideal responses are likely to have high Smarts Ratings. After taking into account all the information at her disposal — previously observed responses and previously conceived ideal answers — she hazards a guess.

The task implicit in the Guess phase is best clarified in terms independent from the Luna Game. The goal is to learn a mapping from ordered sets of natural language samples to a bounded subset of the real numbers that minimizes (L_1) error on test data. Given the inspiration of this problem, I call it the Luna Rating Prediction (LRP) problem. In this chapter, I present a collection of simple experiments to establish a baseline for the problem. I conclude with a discussion of avenues for future work and an argument for LRP's general applicability beyond the scope of Luna.

LRP avoids some of the difficulties of natural language understanding, but also introduces a host of new challenges. For example, it is possible in principal to achieve very good results on LRP without a complete understanding of the natural language responses being analyzed. On the other hand, a complete understanding of the responses is not sufficient to solve LRP, since ratings must still be predicted. Machines may excel at mapping representations of responses to ratings, while humans may excel at representing the responses. A method for LRP must excel at both tasks.

5.1.2 Related Work

LRP may be viewed as a relaxed version of a problem that has been previously considered: Automatic Short Answer Grading (ASAG) (Burrows et al., 2015, Pulman & Sukkarieh, 2005, Sukkarieh & Blackmore, 2009, Ziai et al., 2012). Motivated by standardized testing, ASAG attempts to grade student responses to a fixed set of questions. While LRP is concerned only with the cumulative rating of all responses for a particular respondent, ASAG requires scores for each question. If player ratings are assumed to be completely determined based on their responses to the given set of questions, then LRP reduces to ASAG. In this scenario, a solution to ASAG would imply a solution to LRP, but LRP may be solved without solving ASAG. For example, if one question is much more predictive of a respondent's overall rating than other questions, LRP would be able to leverage this correspondence, while ASAG would be forced to grade all questions, each contributing equally to overall error.

Previous work on ASAG provides a baseline for work on LRP. A recent review by Burrows et al. summarizes advances in ASAG, drawing upon over 80 papers with 35 distinct systems (Burrows et al., 2015). These systems differ in their natural language processing techniques, model building, grading models, and effectiveness. Each of these systems could be potentially adapted for LRP, offering several possible avenues for future research. Here I focus on the state-of-the-art system developed by Education Testing Services (Heilman & Madnani, 2013). This system serves both as a representative example of ASAG systems and as a starting point for LRP.

The latest ASAG system created by ETS performs logistic regression to classify responses as correct or incorrect based on two categories of features: word and character n-gram features, and text similarly features between response and answer key or response and other responses. The system also groups questions into different problem domains and then uses a "domain adaptation" technique, which introduces three separate copies of each feature for generic, domain-specific, and question-specific weighting. This straightforward logistic regression approach outperformed all other existing techniques on the two tested datasets. The general approach of defining a wide variety of natural language features and training a regression model is common among competing systems.

5.2 Methods

5.2.1 Datasets

I use two datasets to evaluate various approaches to LRP. Both datasets are comprised of responses to standardized tests with short answer questions. The first dataset was introduced by Basu et al. (2013) in their work on computer-assisted grading (Basu et al., 2013). They created the dataset by posing 20 questions from the 2012 United States Citizenship Exam to workers on Amazon Mechnical Turk, collecting a total of 698 responses. 10 of the questions were selected for grading by three human judges, who marked each answer as each correct or incorrect. I refer to this dataset as the "Powergrading" dataset, derived from the name of the system that Basu et al. developed.

The second dataset I use for LRP was introduced by Mohler et al. (2011) in their work on ASAG (Mohler et al., 2011). These data consist of responses to short answer problem set and exam questions in a Data Structures course at the University of North Texas. In total, the dataset contains 24 complete responses to 87 questions. Responses are graded by two human judges on a scale from 0 to 5, and these two grades are averaged together per question. I refer to this dataset as the "Mohler 11" dataset. Responses in this dataset are typically longer and more involved than responses in the Powergrading dataset. This level of natural language, combined with the relatively small number of samples, makes the Mohler dataset significantly more challenging for LRP than the Powergrading dataset.

Neither of these datasets are immediately in the format required for LRP. In particular, each question is graded individually, as opposed to each respondent possessing an overall rating. To convert to LRP, I simply add all grades for each respondent and discard the per-question grades. Next I normalize these overall respondent ratings so that all ratings are on a scale of 0 to 1. The final rating preprocessing step is splitting into training and test sets at a ratio of 4 to 1 respectively. To preprocess text responses, I remove all punctuation, stop words, and convert all remaining characters to lowercase.

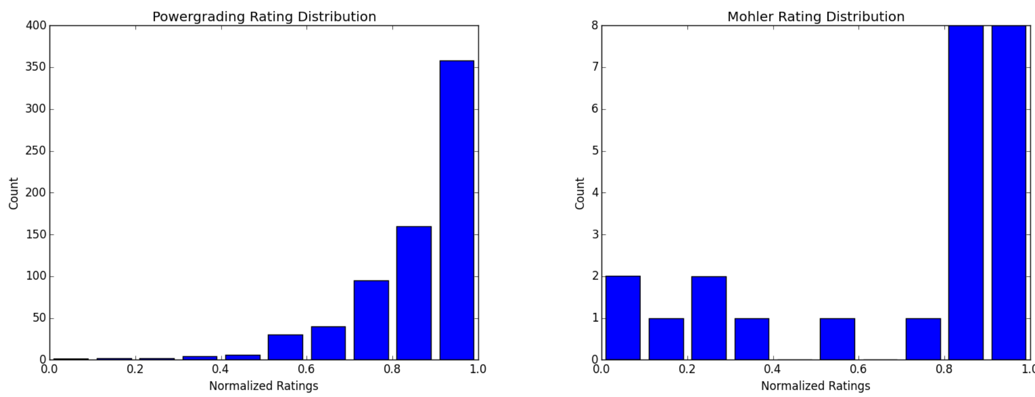


Figure 5.1: Normalized ratings for two datasets used for evaluating approaches to the Luna Prediction Rating problem. The Powergrading dataset, comprised of responses to a subset of the 2012 United States Citizenship Exam, was introduced by Basu et al. (2013) (Basu et al., 2013). The Mohler data, comprised of responses to problem set and exam questions in a University of North Texas Data Structures course, was introduced by Mohler et al. (2011) (Mohler et al., 2011). Ratings were computed by summing individual question grades and normalizing.

5.2.2 Features

Beyond stating LRP as a new problem for the machine learning community, the main contribution of this chapter is a thorough exploration of approaches to the problem. Following the recent work done on ASAG, my strategy is to associate several features with each set of responses, which can be subsequently used for regression. I divide the feature definitions into two categories: question-based features and response-based features. The former takes into account expected responses, while the latter focuses more on data provided by previous rated respondents. I then experiment with several types of regression to combine the features for an optimal model.

Question-Based Features

I assume now that Luna Game players have a set of expected answers for every question, which I refer to as an "answer key". The first time that the Luna Game is played with these questions, the player's only option is to compare the opponent's responses with the answer key. The closer a response is to the expected response, the higher the player's guess of their rating will likely be. Framed in this way, the problem becomes one of measuring the similarity between two samples of natural language.

Measuring semantic similarity between two samples of natural language can be a difficult task. If the two sentences differ only in word order, or only in a few words, then simple approaches can reliably detect similarity. By considering synonyms of the words in both sentences, the similarity measure may be improved. However, in many cases, it is impossible to accurately capture similarity without a deeper semantic understanding of the two sentences. One common approach for representing semantic similarities is to infer vector representations for words in the vocabulary of the training corpus, and then to calculate similarity based on some geometric measure, such as cosine similarity. The problem of inferring these vectors, referred to as neural embeddings, is an active area of research.

In the list below, I present all question-based features considered. In all cases, if there are multiple answers in the answer key, I take the maximum value over all possible values. All question-based features result in a vector whose length is equal to the number of questions in the dataset.

1. Binary Word Overlap (BWO): The i^{th} entry of this feature vector is 1 if any words in the i^{th} response appear in the answer key and 0 otherwise.

2. Fraction Word Overlap (FWO): The i^{th} entry of this feature vector is the number of overlapping words in the i^{th} response with the corresponding answer key, divided by the number of words in either the response or the answer key.

3. Character Edit Distance (CED): The i^{th} entry of this feature vector corresponds to the edit distance (Levenshtein distance) between the i^{th} response and the corresponding answer key at the level of characters. The value is 1.0 minus the fraction of the edit distance and the length of the larger string.

4. Word Edit Distance (WED): The i^{th} entry of this feature vector corresponds to the edit distance (Levenshtein distance) between the i^{th} response and the corresponding answer key at the level of words. For example, the WED between "hello, world" and "goodbye, moon" is 2. The value is 1.0 minus the fraction of the edit distance and the greater number of words in either string.

5. Word2Vec Cosine Similarity (WCS): First I train a neural embedding of words using the answer key as a corpus. Then the i^{th} response and the corresponding answer key are compared, and the i^{th} entry of the feature vector is the maximum cosine similarity between the two. I use the library provided by Gensim here (Sojka, 2010).

6. Semantic Nets (Li et al.) (SNL): For a more sophisticated approach to short sentence similarity, I refer to the work of Li et al. (2006) (Li et al., 2006). They offer a comprehensive metric that takes into account lexical, semantic, and syntactical information about the two sentences being compared. Their system is trained on a corpora of English sentences separate from the training data. I use an existing Python implementation of SNL that uses the bird2009natural English corpora (Bird et al., 2009). The output of their similarity measure is a value between 0 and 1. Thus the i^{th} entry of this feature vector contains the SNL similarity measure between the i^{th} response and the corresponding answer key. One notable downside to this algorithm is that it takes significantly more time (up to 20x) than any other feature described in this chapter.

One inherent challenge in the design of question-based features is the limited size of the answer key. At best, an answer key will document several responses that are considered completely correct. Unfortunately, for open-ended questions, the number of possible completely correct responses can be unbounded. Writing all possible correct responses becomes impractical and many responses may be incorrectly processed due to superficial differences with the answer key. Here I suggest two possible approaches to mitigating this problem. First, the answer key may be multiplied by applying various transformations, such as considering the synonyms of words in the original answer key or performing rearrangements in the syntax of the sentence. Second, as the training set increases, one may choose to include the answers of highly rated players in the answer key. In both cases, one should be sure that growing the answer key does not add prohibitive computational cost.

For future work on developing question-based features for LRP, it may be worthwhile to adapt methods for two related problems: paraphrase detection and textual entailment. Paraphrase detection assesses whether one sample is a paraphrase of the other. Textual entailment is the task of detecting whether one sample of natural language logically implies another sample. For our purposes, I might like to give high ratings to players whose responses imply the answer key, even if they are not exactly the same. Both of these problems are active areas of research in natural language processing and new results will likely be relevant to LRP as well.

Response-Based Features

As a player asks the same questions over several rounds, learning the opponent's true rating at the end of each round, the player's rating model should be updated to take into account the new data. For example, if a new set of responses is very similar to those of a previous opponent, the player's guess of the new opponent's rating should be close to the known rating of the previous opponent. Here I ignore the answer key and consider features that may be extracted solely from responses. In the list below, I present all response-based features tested.

1. Bag of Words (BOW): My simplest response-based feature is the standard natural language processing technique of Bag of Words. Given all the training data, I begin by creating a shared vocabulary list of all words that appear in any responses. I then associate a binary vector with each response in the training data, where entry i is 1 if the response contains the i^{th} word in the vocabulary and 0 otherwise.

2. Bigrams (BIG): This feature is the same as BOW, except rather than building a vocabulary of single words, I build a vocabulary of two adjacent words.

3. Bag of Synsets (SYN): This feature is similar to BOW. I create a thesaurus for every word in the generated vocabulary. I use the WordNet database for finding synonyms (Fellbaum, 1998). Then I associate a vector for each response in the training data, where entry i is 1 if the response contains any synonyms (including the word itself) of the i^{th} word in the vocabulary.

4. Nearest Neighbor via Overlap (NNO): This feature is based on the premise that similar responses will result in similar contributions to the respondents' ratings. For a given response, the most similar response in the training set is identified according to the number of overlapping words. The rating of the corresponding respondent is put into the feature vector, so that the final output is a vector containing the ratings of the most similar respondents per question.

5. Nearest Neighbor via Character Edit Distance (NNC): This feature is the same as NNO, except similarity is defined by character edit distance rather than overlap.

6. Nearest Neighbor via Word Edit Distance (NNW): This feature is the same as NNO, except similarity is defined by word edit distance rather than overlap.

5.2.3 Regression

With features defined, the task then takes the standard form of a regression problem. For simplicity, I assume here that the rating is a linear combination of the features. I then consider two models of regression. The first is Ordinary Least Squares Linear Regression, which minimizes the sum of squares between predicted and actual ratings. I then run Lasso Regression ($\alpha = 0.1$), which biases towards simpler models by penalizing the number of nonnegative coefficients in favor of sparse solutions. In both cases, I use the implementations in the Scikit-Learn library for machine learning (Pedregosa et al., 2011).

5.3 Results

I ran OLS Linear Regressions and Lasso Regressions on each of the 12 features for 100 trials. In addition, I included a combination of all 12 features (COM), and two controls. The first control predicts a random rating for a given set of responses (RAN). The second control, which I refer to as the "constant best guess", always predicts the average of all previously seen ratings (CBG). To consider a feature useful, it should at least outperform both of these controls on average. I define performance in terms of the Root Mean Square Error between predicted ratings and actual ratings in the test set. The results of these experiments are depicted in Figure 2. Lasso Regression yielded no meaningful differences from OLS, so I depict only the latter.

The relative performances of the approaches seem to be consistent between the two datasets tested. As expected, the Mohler 11 dataset proves far more difficult, likely due to the very limited size of the training set, the large number of responses per respondent, and the longer length of each response. Also unsurprising is the result that for OLS Linear Regression, a combination of all features outperforms any single feature. The combination does not perform as comparably well in Lasso Regression, presumably because the number of features actually included is small due to the penalty term. In both cases, any margin of improvement offered by a combination of all features is slim. Simple response-based features such as Bag of Words or Nearest Neighbor via Overlap are competitive with the combination. In practice, it may be preferable to use one or both of these due to the savings in computa-

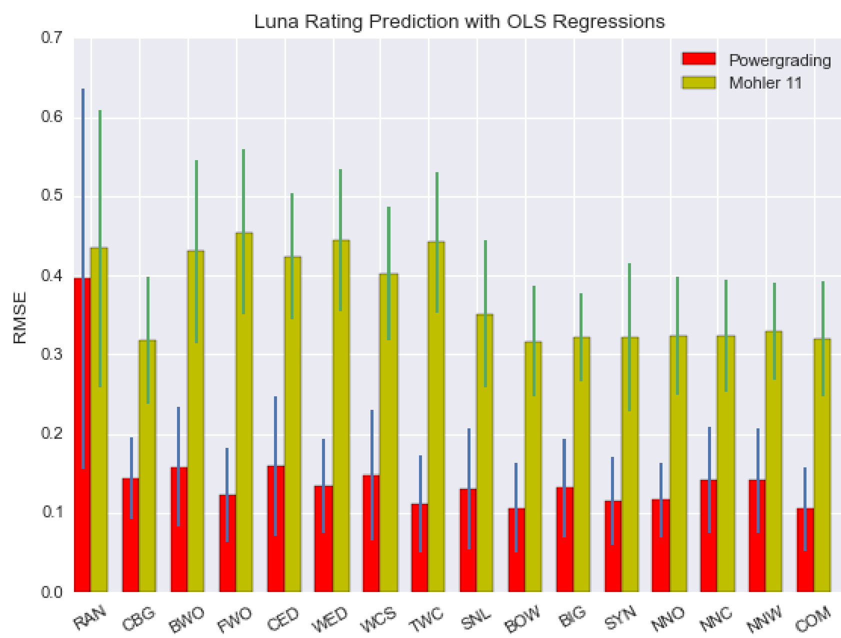


Figure 5.2: Fifteen approaches to the Luna Prediction Rating problem on two datasets using Ordinary Least Squares Linear Regression. From the left, the first two approaches represent baseline controls; the following six approaches are each OLS Linear Regressions on single question-based features; the following six are also OLS Linear Regressions, but on single response-based features instead; the last is an OLS Linear Regression that includes all 12 previous features. The Root Mean Square Error averaged over 100 trials is plotted for each approach. Refer to the text for the feature abbreviations.

tional resources. Among the question-based features, Binary Word Overlap fares the best, but cannot outperform the Constant Best Guess baseline. In future work, the question-based features could likely be improved by augmenting the answer keys as discussed above.

5.4 Discussion

In this chapter, I defined the Luna Rating Prediction problem of guessing a respondent's rating based on previous observations of responses and ratings. I discussed the closely related problem of Automatic Short Answer Grading and demonstrated how datasets and methods for ASAG can be adapted for LRP. I then presented initial work on LRP, which centered around several features of response sets that I use to learn a model for predicting ratings. These features were presented in two categories: question-based features, which rely only on a preset answer key, and response-based features, which ignore the answer key but take advantage of seen responses from previous rated respondents. I combined these features using traditional regression techniques to arrive at a comprehensive system for LRP.

For a player of the Luna Game, the ability to solve LRP is critical. The outcome of a Game hinges on the difference between predicted ratings and actual ratings. A human player may take advantage of the machine learning approaches to LRP offered here, but more likely the human will learn to rate well after several rounds of play without any formalisms or explicit machine learning. This human ability is based on an understanding of natural language, a deep knowledge base, and the relative ease with which humans judge one another in everyday scenarios. As machine players build an understanding of natural language and a knowledge base from answering questions in the Luna Game, these advances should be transferrable to LRP as well. A truly intelligent player of the Luna Game is unlikely to consider LRP and question answering as completely separate problems, but rather as manifestations of one general underlying problem.

Throughout this chapter, I sought to treat LRP as a general problem beyond the scope of the Luna Game, since I expect work on the problem to have several applications. Any situation in which a judge wishes to automatically evaluate a respondent can be thought of as an instance of LRP. The quantity being evaluated may be completely known to the respondent, as is the case in the Luna Game. However, in some applications, the quantity may be only partially known or unknown. For example, consider the case of a government official wishing to estimate the probability that a convict will repeat an offense based on questionnaire responses, or the case of an health insurance company wishing the estimate the amount of money that a customer will require for medical care. These real life applications combined with the motivation provided by the Luna Game make LRP a problem worthy of further study.

I rarely find it useful to distinguish between theory and practice; their interplay is already profound and will only increase as the systems and problems we consider grow more complex.

Michael I. Jordan

6

A Web Implementation of the Luna Rating System

6.1 Introduction

The Luna Rating System is designed to be a practical test for machine intelligence. It may offer some utility as a thought experiment, but its primary value can only be realized in practice. In this chapter, I describe the design and launch of the first web-based implementation of LRS. The application is meant to serve as a comprehensive proof-of-concept. For the application to fulfill its purpose, it must first be capable of recruiting and handling hundreds of human players. The design should then persuade players to play several games, so as to refine their Smarts Ratings. Finally, the behavior of players should indicate a collective understanding of the Luna Game, and evidence should suggest their aspirations towards honest play.

The baseline success of the LRS implementation can be easily measured through traffic statistics: how many players play, and how long do they typically stay? Such statistics comprise the first section of results in this chapter. In addition to these standard metrics, the questions and responses traded by the players can be assessed qualitatively. I provide a representative sample of questions and discuss common themes from the full dataset. The final source of results is derived from human players' interactions with simple standardized machine players. I argue that the Guesses that human players make of the machine players should converge over time as a consensus on the semantics of Smarts Ratings is reached. Machine players also provide a demonstration for AI researchers who wish to introduce their own machines into the system.

6.2 Building the Web Interface

6.2.1 Design of the Web Interface

My implementation of the Luna Rating System is built on the open-source software stack consisting of MongoDB, Express.js, Angular.js, and Node.js, which is collectively referred to as the MEAN stack (Karpov, 2013). The latter three technologies are all written in JavaScript and MongoDB is a NoSQL database. At a very high level, MongoDB stores all player and game data, Angular.js controls and displays the client side, Express.js forms the foundation of the server side, and Node.js runs the code written on top of that foundation. Notable libraries used include: Mongoose.js, a Node.js library for interfacing with MongoDB; Passport.js, a Node.js library which provides middleware for user authentication; and Angular-UI, an Angular.js library that simplifies routing, i.e. navigation between different states of the website. The front end of the website, i.e. the style, formatting, and organization of content, was modified from the "Material Admin" LESS/Angular.js template, for which I purchased a single application license. The site was developed locally and then hosted on EC2 by Amazon Web Services under the domain name "luna-game.com".

The website?s wireframe layout was first prototyped on paper, which allowed for iteration on the skeletal design (Rettig, 1994). The centerpiece of the website is the ongoing Luna Games, and a secondary component allows users to review their performance in previous Games. To reduce user drop-off, the front end seamlessly leads a user between the phases of a Luna Game within a single page. All opportunities for latency — between phases of a Game or between Games — are reduced as much as possible to encourage repeated play. To increase the probability that a first-time user explores the website beyond the landing page, the website has an option to play as a "guest." The only functional difference between a user that signs up and a guest is that signed up users may log out and log back in, while a guest account expires once the 60-day token is removed from the browser or the user explicitly signs out. Further consideration was given to user experience in the context of the results presented in Chapter 3 of this thesis, which demonstrate that all players must have a clear understanding of their goals during a Luna Game. While it is important that users be able to start playing Luna Games as quickly as possible, they should first be fully aware of the Game's structure and their goals.

Weighing these considerations, the core features of the website are divided into 10 primary states and 4 secondary states. The secondary states include static pages — the contact page and two extended information pages — and one profile page that allows a user to review statistics related to her historical performance. The primary states are presented in Figure 6.1. They include four states corresponding to the phases of a Luna game, one home state, which contains games as nested states, a landing page, a guest state, a sign up state, a login state, and an information state. Note that first-time users must provide consent and review information about the structure of Luna Games before they are able to play. The eager first-time user is able to reach a new game with only three clicks from the home page, and a returning user can reach games with two clicks and a form submission. This simple state architecture ensures that users are well-informed of the rules and then playing as soon as possible. Screenshots of a typical user experience are depicted in Figure 6.2.

6.2.2 An API for Machine Players

To allow AI researchers to enter machines as players on the Luna Rating System website, I implemented a RESTful API. Machine players must first register separately from human players to receive an API token. This process is currently done by hand, since the system is not yet protected against malicious machine players (e.g. there is no limit on the number of API calls that one player may make). The experience of machine players is very similar to that of humans. The machine may make a request to start a new game, provide questions during the interview phase of an ongoing game, receive questions and provide answers during the response phase, provide a guess during the guess phase, and receive any updates of its own Smarts Rating and total wins. Notably, the guess of a machine is not factored into the Smarts Rating of its opponent. A Python template, which may be used directly or as a

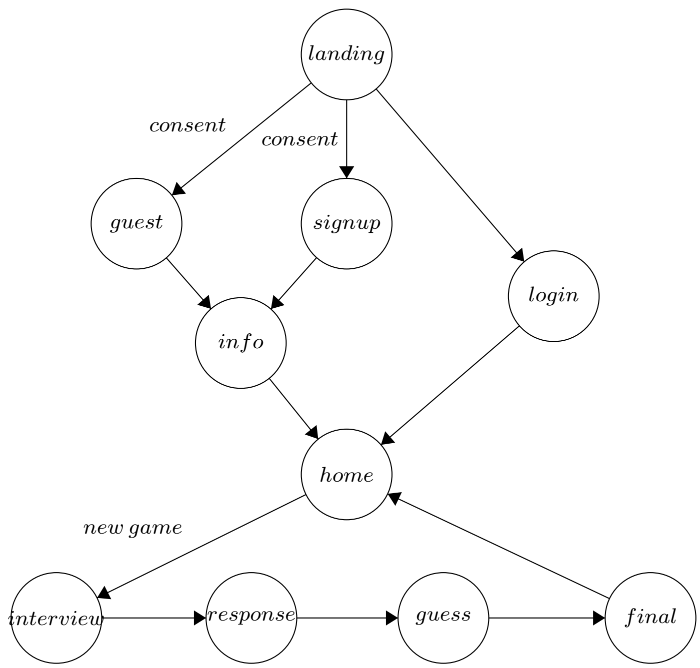


Figure 6.1: Schematic diagram depicting the primary states of the Luna Rating System web interface. Nodes represent website states and edges indicate how users typically navigate between states. A first-time user arrives at the *landing* state. Upon providing consent, the user may play as a *guest* or *sign up*, both which lead to the *info* state. The info state explains the Luna Game and then directs users to the *home* state. The user can then create a new Luna Game, launching the *interview* state. As the user progresses through the Luna Game, she transitions from interview to *response*, to *guess*, and to *final*. They then return to the home state to start another game. Returning signed up users may *login* from the landing state to reach the home state.

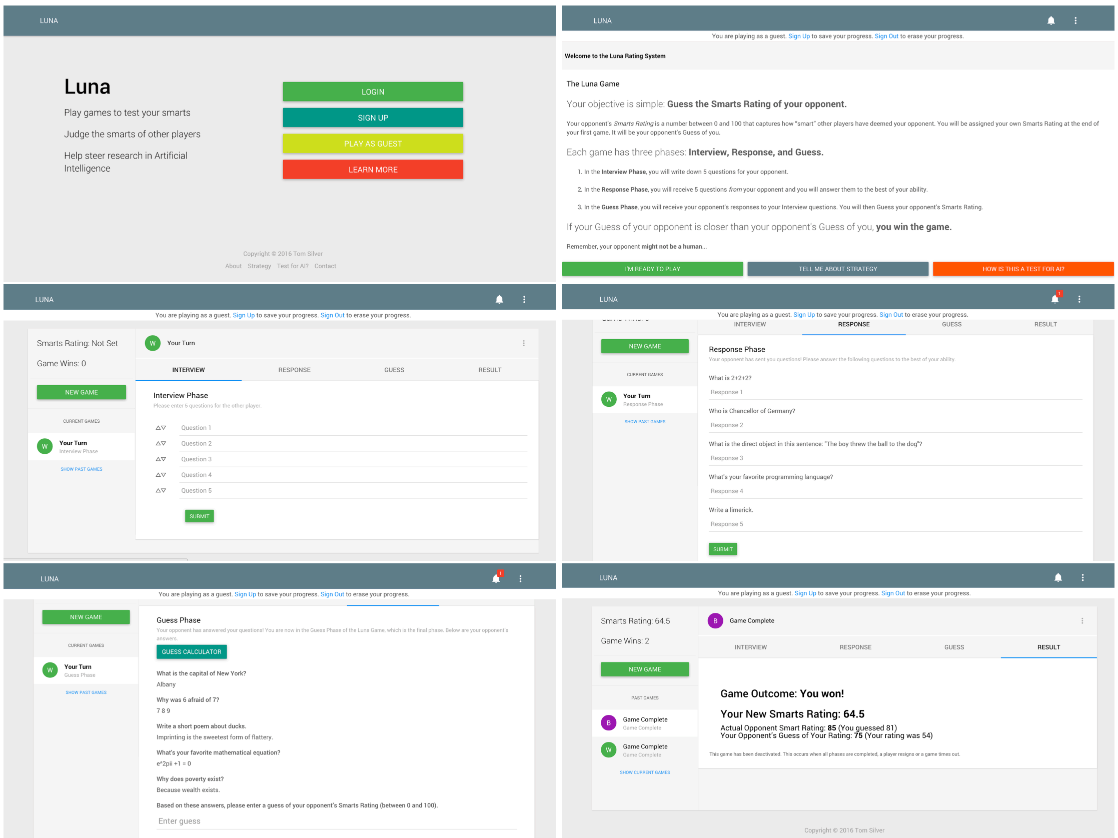


Figure 6.2: Exemplary screenshots of the Luna website. Clockwise from top left: the landing page, the information page for first-time users, interview phase, response phase, guess phase, and final phase.

blueprint for other languages, is provided for researchers.

This study implemented two simple machine players, both of which were equipped with a fixed set of 5 questions to present during all Interview Phases for consistency. The questions were:

1. What color is the sky?
2. What is the direct object in this sentence: “The boy threw the ball to the dog”?
3. Why is 6 afraid of 7?
4. Why does poverty exist?
5. What is the capital of New York?

The only distinction between the two machines was their response functions. The first machine, which I refer to as the Gibberish Bot, responds to any question by generating a random string of uppercase characters, digits, and spaces. The length of the response is also randomly selected to be between 1 and 200. This machine is meant to serve as the most basic of controls; if Smarts Ratings are at all meaningful, the Gibberish Bot should have a very low one. The second machine, referred to as the Cleverbot, responds to questions by querying the Cleverbot chatbot, a web-based chatbot that responds to human inputs by outputting previously seen human inputs that are deemed most similar according to a simple surface-level analysis (Carpenter, 2015). Cleverbot is an ideal second control, as it provides responses that have been written by humans, but does not attempt to understand these questions with any level of sophistication. Thus one would reasonably expect Cleverbot to achieve a Smarts Rating higher than the Gibberish Bot, but lower than most humans. Both bots played 9 games, alternating one at a time against randomly selected human opponents in the first few days of the launch.

6.2.3 Launching the Website

Since this work relies on human participation, the study was submitted for review to the Committee on the Use of Human Subjects in Research at Harvard University. The study qualified for expedited review and received approval on February 18, 2016. The website was immediately launched and publicized, primarily via social media and email. Publicity efforts continued for one week following the launch. The results presented in this chapter were gathered in the two weeks following launch, up until March 3, 2016.

6.3 Results

6.3.1 Traffic Summary

In the first two weeks of the Luna web implementation, the site attracted 1,293 registered players (including the two machine players). 685 Luna Games were initiated, creating 4,173 unique questions and 4,923 responses. According to Google Analytics, the website earned 2,312 sessions and 2,937 page views. Slightly over 30% of visitors were returning to the site. 54% of the visitors came to the site directly, e.g. through clicking a link in their email or typing the URL manually, while the remaining visitors were referred primarily through Facebook and Twitter. The daily number of sessions is plotted in Figure 6.3. As expected, the amount of traffic correlates strongly with publicity efforts, which were concentrated in the first few days of the launch and virtually ceased after one week.

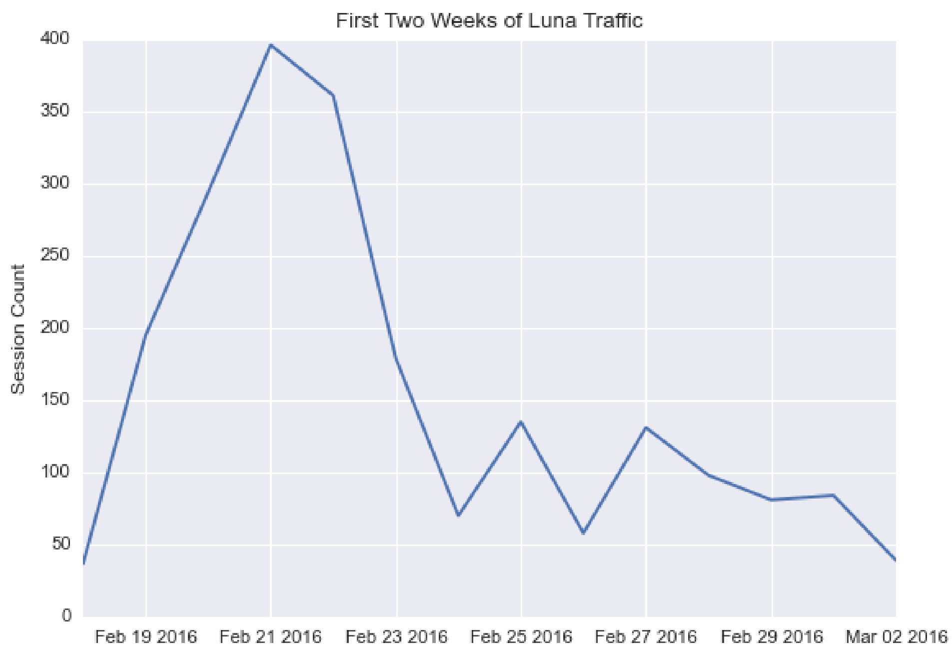


Figure 6.3: Daily traffic to the Luna web implementation in the first two weeks after launch. Publicity for the site was concentrated in the first few days of the launch and virtually ceased after one week.

6.3.2 Human Player Summary

The vast majority of human players registered as guests (1,207), while very few (86) signed up using their emails. One consequence of this discrepancy is that only a small fraction of users were able to receive email updates when it was their turn to play in a Luna Game. In addition, many users appear to have registered as guests and then left the site without submitting interview questions; the overall average number of games played was 0.33, with a median of 0. This is not to say that the guest registration option was ill-advised; it is likely that removing the guest option would have yielded far fewer registrations overall. However, anecdotal evidence suggests that the frequent stalling of Luna Games, due to guests leaving in the middle of a Game, led to frustration among more dedicated players. Among the 220 players who did complete at least one full game, the average Smarts Rating was 68.9, with a median of 74. The full distribution of Smarts Ratings for human players is depicted in Figure 6.4. The ratings exhibit a bell curve roughly around the mean with a few low outliers.

Sample Questions and Responses

I plan to publicly release the complete dataset of questions and responses through the Harvard Dataverse Network after the submission of this thesis. The below 10 examples are meant to be representative of the full dataset in terms of question and response length, form, and content.

1. Q: Is addiction a disease, as is increasingly being claimed, or is the addict at fault for developing their addiction?

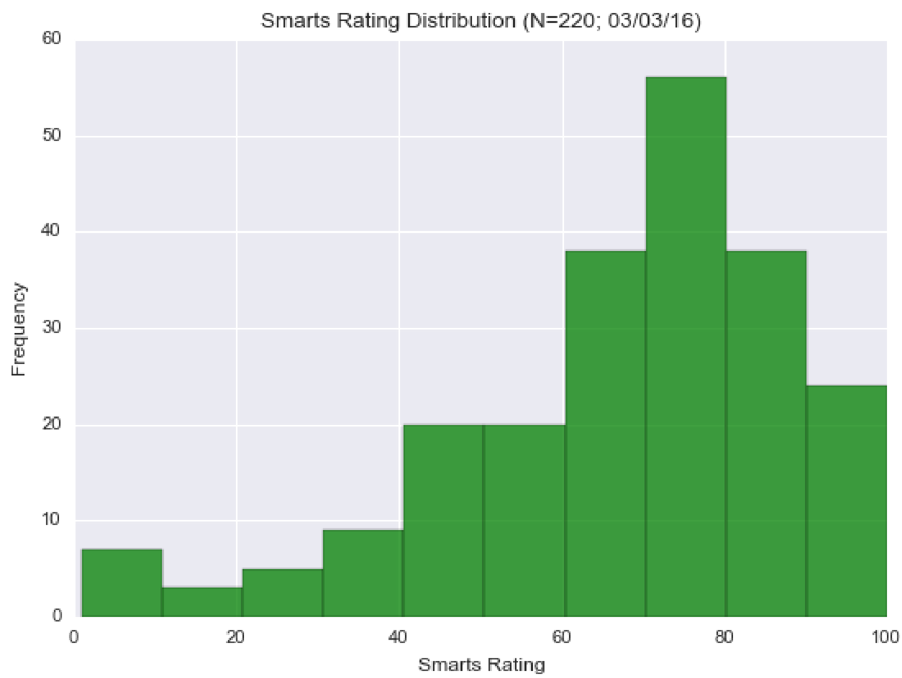


Figure 6.4: Smarts Rating distribution of human players who completed at least one full Luna Game. The ratings exhibit a bell curve roughly around the mean of 68.9 with a few low outliers.

A: a bit of each; people can be psychologically and/or genetically liable to develop addictions, but with sufficient willpower they needn't

2. Q: Apparently, our senses can detect around 11 million pieces of information per second. How much total can a human process of this 11 million? A: 10 of those B: all 11 million C: 25 to 30 of those D: 5 to 7 of those

   A: B

3. Q: Compose a limerick involving the word jello.

   A: hey there fellow; I hate jello; it's not tasty; I'm being hasty; hard question, well-oh...

4. Q: What is the biggest risk from self-driving cars?

   A: Two things. First, we currently have no way of integrating self-driving cars into current traffic patterns. The self-driving cars are better at driving than people are, and a road with only self-driving cars would be much safer and far more efficient, but trying to work around humans will cause all sorts of headaches and loss of efficiency. For example, there are algorithms for having autonomous cars coordinate at interesections so that they can all drive through a full speed from all directions and never hit each other; if you had to throw even one human-driven car into that mix, then whole system would have to stop. Second, current cars are disturbingly hackable. Security researchers have demonstrated over and over that it's super easy to remotely take control of a car, and have the ability to mess with it in all kinds of ways (e.g. disable the breaks), but car companies just don't care and will not implement any security because consumers don't care and it would cost the companies extra money. Obviously as cars become even more autonomous, this security risk will grow even further. Sorry, that was really long... but I spend too much time thinking about these things.

5. Q: What causes the McGurk effect?

   A: Our brain confuses visual inputs with auditory ones.

6. Q: What is the strangest experience you have ever had while riding a bus?

   A: Overhearing a taxidermist telling a random person how much he loved his job and then asking how their pet was doing

7. Q: Who am I?

   A: Is a question that usually strikes us with the most weight around the age of 50.

8. Q: Should Supreme Court positions have term limits?

   A: don't understand the question. is this about USA? this game is worldwide.

9. Q: Who is/was the most influencial person in your life?

   A: I am

10. Q: Describe an algorithm for walking

    A: Brain signal -> knee. ˆknee. Increment 37 degrees. +1.5 feet forward. Repeat for other knee.

6.3.3 Machine Player Summary

After 9 games each, the Gibberish Bot achieved a Smarts Rating of 6.4, and Cleverbot arrived at a Smarts Rating of 40.4. All opponents but one guessed a Smarts Rating of 1 for the Gibberish Bot; the remaining opponent guessed 50. Guesses of Cleverbot's Smarts Rating ranged from 5 to 88. To illustrate the typical behavior of each of these machines during the Response phase, I provide representative samples below.

Gibberish Bot Response Phase Sample

1. Q: How old are you?

   A: WE23 F1MTR4FI7YBT16JYRV8AWEY7ALLLXI3M79 1BSL8G

2. Q: Will your answer to this question be wrong?

   A: MGKGJNYJYHP4XM2J35TQ BF4GNNERA77Z20ZVULH

3. Q: Is the space L$\hat{2}$ complete?

   A: DZTYWUKXE4D13B58A823VA8 22ZLZ19FQAGPEMQ

4. Q: What is a contradiction to the CAPM model of financial markets?

   A: PGIITGJWK0NFBTOT

5. Q: What is a bound on spectral sparsification of complete graphs?

   A: NS2TEEMRSL91ANNF 4LDK2M5YSI4I1JVU6SV74LUKNVAH

Guess: 1

Cleverbot Response Phase Sample

1. Q: What is your favorite book and please describe it in 3 words.

   A: Howl by Allen Ginsberg.

2. Q: What's the best thing you did today?

   A: The moment that I fell in love was the best thing I have seen. You?

3. Q: What is your favorite number from 1 to 10 and why?

   A: Number 13 is my favorite number.

4. Q: What is the best machine ever invented?

   A: Back to the Future is the best movie ever.

5. Q: What is your biggest accomplishment of the last year?

   A: All the time I HAVEN'T spent talking to you!

Guess: 40

6.3.4 Game Summary

Among the 685 initiated Luna Games, 292 were followed through to the final phase by both players. To see if there is any obvious correlation between the length of responses and the subsequent guess of the Smarts Rating of the responder, I created a scatterplot of response length and subsequent guess (Figure 6.5). The relationship between response length and guess gives a Pearson's r of 0.179, suggesting a weak positive correlation. One interpretation of this result is that players who put more effort into their responses are rewarded with higher Smarts Ratings.

It is impossible to assess whether Smarts Ratings are accurate without presuming some ground truth about intelligence. However, one encouraging sign would be the convergence of a player's Smarts Rating as she continues to play games. In Figure 6.6, I plot the standard deviations of Smarts Ratings per player as a function of the number of games each has completed. No such convergence is yet observed, likely indicating that more playing time is necessary for the system to reach an equilibrium in which most players hold a shared understanding of Smarts Ratings.

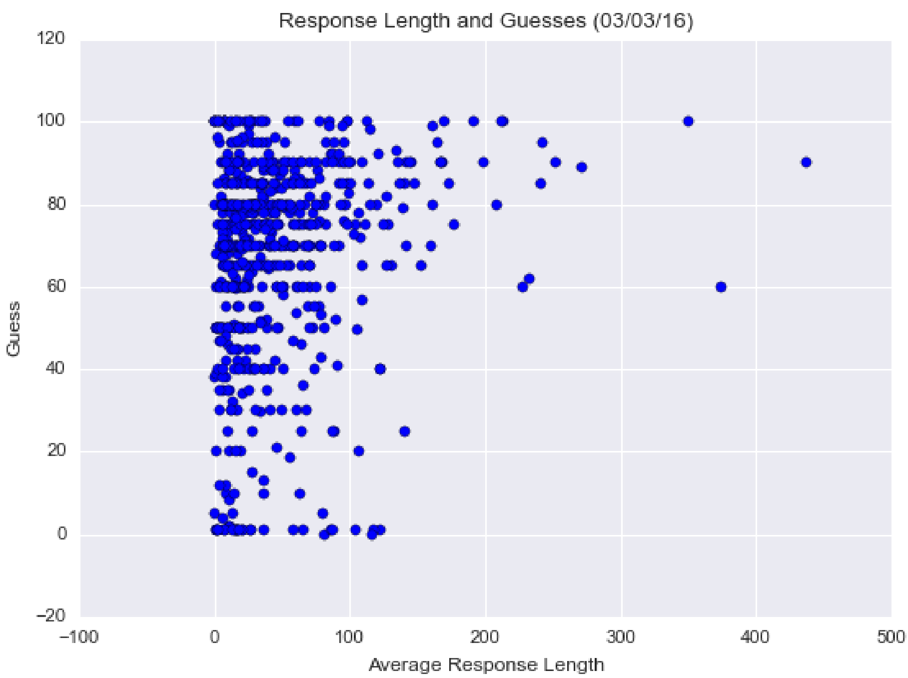


Figure 6.5: Relationship between response length and subsequent guess of the responder's Smarts Rating. Pearson's r is 0.179, suggesting a weak positive correlation.

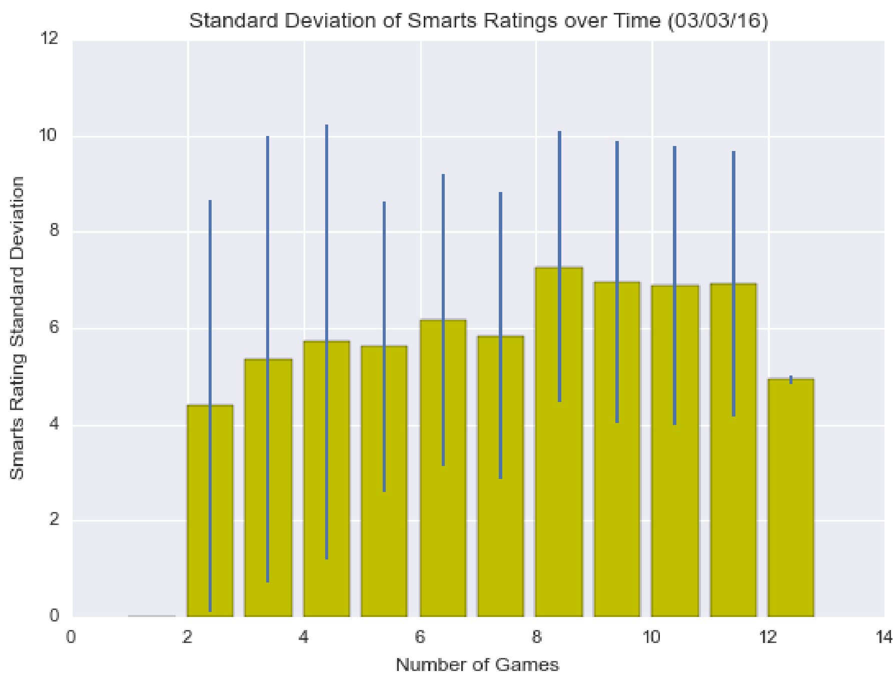


Figure 6.6: Standard deviations of Smarts Ratings per player as a function of number of games completed. Convergence as the number of games increases is not yet observed. Note that the values in the plot are standard deviations, and the error bars are standard deviations (over all players) of standard deviations.

6.4 Analysis

The first goal of this chapter — to demonstrate that a straightforward web implementation of the Luna Rating System can draw players and sustain games — is readily reached. Given the relatively minor publicity efforts, unaided by any sort of financial expenditures, the substantial traffic to the website within the short time window is very encouraging. The Smarts Ratings reached by both human and machine players are also auspicious. The bell curve observed for humans is consistent with the distribution of IQ, for example. Moreover, the Smarts Ratings of the machines — 6.4 for the Gibberish Bot and 40.4 for Cleverbot — indicate that the controls were relatively successful. As expected, the Gibberish Bot reached a much lower rating than Cleverbot, and Cleverbot remained significantly lower than the majority of human players. These results corroborate the semantic value of Smarts Ratings and suggest that human players understand the basic notion behind the ratings.

While the machine Smarts Ratings align with expectations, the extent to which ratings can meaningfully distinguish between human players is less clear. The weak correlation between response length and subsequent guess can be interpreted in two ways. Optimistically, one would expect that players who are more dedicated to demonstrating intelligence might write longer responses, thus deserving higher Smarts Ratings. Pessimistically, one could argue that Smarts Ratings are just a measure of response length, rather than intelligence. Future work could distinguish between these interpretations by enforcing a stricter limit on response lengths, or by attempting to correlate Smarts Ratings with other metrics of intelligence like IQ. Additionally, the lack of convergence of Smarts Ratings suggests that more time is needed for the system to reach an equilibrium. Once such convergence is observed, the discriminative power of Smarts Ratings will be more clear.

The qualitative results presented in this chapter, consisting mainly of the questions and responses of human players, are perhaps the most encouraging. The questions exhibit a wide range of form, content, length, and difficulty. Some questions are yes/no, multiple choice, or fill in the blank, while others anticipate lengthy free form responses. Some questions are straightforward trivia or simple and subjective (e.g. “How are you?”), but many require high levels of understanding and logical reasoning. For example, many questions resemble open-ended prompts that one might expect to see on a standardized writing test (e.g. “Do schools put too much or too little time into Phys. Ed. classes?”). The subsequent responses to these questions are equally broad. The full dataset offers intriguing insight into how humans define and assess intelligence. Moreover, a machine which is to be called artificially intelligent must be able to adequately answer the collected questions.

6.5 Discussion

The results reported in this chapter demonstrate the promising potential of the Luna Rating System as an active and informative benchmark for AI. The study also highlights critical design considerations for future implementations. Perhaps the most confounding aspect of this web implementation was the prevalence of stalled Luna Games. A game stalls when one of the players leaves the website in the middle of gameplay and does not return. The remaining player is left waiting until the game expires due to inactivity (which currently occurs after 3 days). While waiting players are welcome to start simultaneous games, it is likely discouraging to see a backlog of unfinished games; the incentive to start a new game diminishes as the expectation that games will be finished decreases. If a player registers with an email address, the likelihood that she will return to games improves, as the system automatically sends an email to notify her when it is her turn to play. Thus a central objective in future design should be to obtain valid email addresses. Other ways to further persuade players to stay and play should also be considered.

The primary question left unanswered in this chapter is: what levels of participation are required for Smarts Ratings to reach an equilibrium? There are two dimensions of participations that likely affect this convergence: total number of players, and duration of play. I suspect that increasing the duration of play, i.e. the number of games played per player, is more important than increasing the number of players. An individual player develops an understanding of Smarts Ratings over time as she finishes games and true Smarts Ratings are revealed. The majority of players in this first web implementation have finished no games at all, meaning that they have no empirical basis from which to infer ratings. It is quite possible that even a much smaller cohort of players, each of whom plays several games, would reach a Smarts Rating equilibrium. Future efforts should be focused on achieving this equilibrium; the informativeness of the Luna Rating System depends on the precision of Smarts Ratings.

In addition to revealing refinements for future Luna web implementations, this chapter presents an intriguing original dataset that alone offers several clear avenues for future work. There is the social psychology angle: how do people assess one another's intelligence in practice? There is the NLP perspective: what linguistics features of responses are correlated with subsequent guesses? Simply grouping the questions and responses thematically may reveal rich trends. All participants have given consent in which they acknowledge that their questions and responses may be released as part of a publicly available dataset. I plan to compile such a dataset and release it through the Harvard Dataverse Network in the hope of enabling others to conduct their own analyses.

We can only see a short distance ahead, but we can see plenty there that needs to be done.

Alan Turing

7 Conclusion

In this thesis, LRS has been defined, characterized, stress tested, contextualized, and implemented. I have argued that the system offers a practical, accessible, generalizable, continuous, and robust benchmark for AI. I have shown how the Luna Game fits into the broader context of existing AI research and proposed the Luna Rating Prediction problem for future study. I have demonstrated with a web-based proof-of-concept that LRS can recruit and sustain large numbers of human players who ask questions of every variety and answer with depth. But the most important question pertaining to LRS remains: will it be used? The ultimate value of LRS is not something that can be theoretically proved or simulated. As a proposed practical test for AI, its merit can only be determined in practice. The true test of the Luna Rating System is yet to come.

References

Ali, H., Chali, Y., & Hasan, S. (2010). Automatic question generation from sentences: a preliminary approach. In *Proceedings of the Conference on Traitement Automatique de la Langue Naturelle, Montreal, Canada*.

Anderson, J., Baltes, J., & Cheng, C. T. (2011). Robotics competitions as benchmarks for ai research. *The Knowledge Engineering Review*, 26(01), 11–17.

Andrenucci, A. & Sneiders, E. (2005). Automated question answering: Review of the main approaches. In *null* (pp. 514–519).: IEEE.

Androutsopoulos, I., Ritchie, G. D., & Thanisch, P. (1995). Natural language interfaces to databases–an introduction. *Natural language engineering*, 1(01), 29–81.

Aron, J. (2011). How innovative is apple's new voice assistant, siri? *New Scientist*, 212(2836), 24.

Balch, T. & Yanco, H. (2002). Ten years of the aaai mobile robot competition and exhibition. *AI Magazine*, 23(1), 13.

Baltes, J. & Bräunl, T. (2009). Hurocup: General laws of the game 2008.

Basu, S., Jacobs, C., & Vanderwende, L. (2013). Powergrading: a clustering approach to amplify human effort for short answer grading. *Transactions of the Association for Computational Linguistics*, 1, 391–402.

Bird, S., Klein, E., & Loper, E. (2009). *Natural language processing with Python*. " O'Reilly Media, Inc.".

Bishop, M., Shah, H., & Warwick, K. (2010). Testing turing's five minutes, parallel-paired imitation game. *Kybernetes*, 39(3), 449–465.

Block, N. (1980). What intuitions about homunculi don't show. *Behavioral and Brain Sciences*, 3(03), 425–426.

Bollacker, K., Evans, C., Paritosh, P., Sturge, T., & Taylor, J. (2008). Freebase: a collaboratively created graph database for structuring human knowledge. In *Proceedings of the 2008 ACM SIGMOD international conference on Management of data* (pp. 1247–1250).: ACM.

Brill, E., Lin, J. J., Banko, M., Dumais, S. T., Ng, A. Y., et al. (2001). Data-intensive question answering. In *TREC*, volume 56 (pp.90).

Brockman, J. (2015). *What to Think About Machines That Think: Today's Leading Thinkers on the Age of Machine Intelligence*. Edge.org.

Burger, J., Cardie, C., Chaudhri, V., Gaizauskas, R., Harabagiu, S., Israel, D., Jacquemin, C., Lin, C.-Y., Maiorano, S., Miller, G., et al. (2001). Issues, tasks and program structures to roadmap research in question & answering (q&a). In *Document Understanding Conferences Roadmapping Documents* (pp. 1–35).

Burrows, S., Gurevych, I., & Stein, B. (2015). The eras and trends of automatic short answer grading. *International Journal of Artificial Intelligence in Education*, 25(1), 60–117.

Carpenter, R. (2015). Cleverbot [computer program].

Chelba, C., Mikolov, T., Schuster, M., Ge, Q., Brants, T., Koehn, P., & Robinson, T. (2013). One billion word benchmark for measuring progress in statistical language modeling. *arXiv preprint arXiv:1312.3005*.

Collobert, R., Weston, J., Bottou, L., Karlen, M., Kavukcuoglu, K., & Kuksa, P. (2011). Natural language processing (almost) from scratch. *The Journal of Machine Learning Research*, 12, 2493–2537.

Cucerzan, S. & Agichtein, E. (2005). Factoid question answering over unstructured and structured web content. In *TREC*, volume 72 (pp.90).

Dang, H. T., Kelly, D., & Lin, J. J. (2007). Overview of the trec 2007 question answering track. In *TREC*, volume 7 (pp.63).

Fellbaum, C. (1998). *WordNet*. Wiley Online Library.

Ferrucci, D. A. (2012). Introduction to "this is watson". *IBM Journal of Research and Development*, 56(3.4), 1–1.

Gardner, H. (2011). *Frames of mind: The theory of multiple intelligences*. Basic books.

Gates, D. M. (2008). *Automatically generating reading comprehension look-back strategy: Questions from expository texts*. Technical report, DTIC Document.

Genesereth, M., Love, N., & Pell, B. (2005). General game playing: Overview of the aaai competition. *AI magazine*, 26(2), 62.

Glickman, M. E. (1995). Chess rating systems. *American Chess Journal*, 3(59), 102.

Graesser, A. C., McNamara, D. S., & VanLehn, K. (2005). Scaffolding deep comprehension strategies through point&query, autotutor, and istart. *Educational psychologist*, 40(4), 225–234.

Gunderson, K. (1964). Vii.—the imitation game. *Mind*, 73(290), 234–245.

Hayes, J. E. & Levy, D. N. (1976). *The world computer chess championship, Stockholm 1974*. Edinburgh University Press.

Hayes, P. & Ford, K. (1995). Turing test considered harmful. In *IJCAI (1)* (pp. 972–977).

Heilman, M. (2011). *Automatic factual question generation from text*. PhD thesis, Carnegie Mellon University.

Heilman, M. & Madnani, N. (2013). Ets: Domain adaptation and stacking for short answer scoring. In *Proceedings of the 2nd joint conference on lexical and computational semantics*, volume 2 (pp. 275–279).

Heilman, M. & Smith, N. A. (2010a). Extracting simplified statements for factual question generation. In *Proceedings of QG2010: The Third Workshop on Ques-tion Generation* (pp.ĩ1).

Heilman, M. & Smith, N. A. (2010b). Good question! statistical ranking for question generation. In *Human Language Technologies: The 2010 Annual Conference of the North*

American Chapter of the Association for Computational Linguistics (pp. 609–617).: Association for Computational Linguistics.

Hermann, K. M., Kocisky, T., Grefenstette, E., Espeholt, L., Kay, W., Suleyman, M., & Blunsom, P. (2015). Teaching machines to read and comprehend. In *Advances in Neural Information Processing Systems* (pp. 1684–1692).

Hirschman, L. & Gaizauskas, R. (2001). Natural language question answering: the view from here. *natural language engineering*, 7(4), 275–300.

Hussey, M. (2016). Alphago's victory means the world is about to change.

Karpov, V. (2013). The mean stack: Mongodb, expressjs, angularjs and node. js. *Tillgänglig: http://blog. mongodb. org/post/49262866911/the-mean-stack-mongodb-expressjs-angularjs-and [Besökt: 2 juni 2015].*

Kitano, H., Asada, M., Kuniyoshi, Y., Noda, I., & Osawa, E. (1997). Robocup: The robot world cup initiative. In *Proceedings of the first international conference on Autonomous agents* (pp. 340–347).: ACM.

Kolomiyets, O. & Moens, M.-F. (2011). A survey on question answering technology from an information retrieval perspective. *Information Sciences*, 181(24), 5412–5434.

Kunichika, H., Katayama, T., Hirashima, T., & Takeuchi, A. (2004). Automated question generation methods for intelligent english learning systems and its evaluation. In *Proc. of ICCE*.

Kwok, C., Etzioni, O., & Weld, D. S. (2001). Scaling question answering to the web. *ACM Transactions on Information Systems (TOIS)*, 19(3), 242–262.

LeCun, Y., Cortes, C., & Burges, C. J. (1998). The mnist database of handwritten digits.

Legg, S., Hutter, M., et al. (2007). A collection of definitions of intelligence. *Frontiers in Artificial Intelligence and applications*, 157, 17.

Li, Y., McLean, D., Bandar, Z. A., O'shea, J. D., & Crockett, K. (2006). Sentence similarity based on semantic nets and corpus statistics. *Knowledge and Data Engineering, IEEE Transactions on*, 18(8), 1138–1150.

Littman, M. & Zinkevich, M. (2006). The 2006 aaai computer poker competition. *ICGA Journal*, 29(3), 166.

Loebner, H. (2003). Home page of the loebner prize-the first turing test. *Online unter http://www. loebner. net/Prizef/loebner-prize. html.*

Mahoney, M. (2006). Rationale for a large text compression benchmark. *Retrieved (Aug. 20th, 2006) from: http://cs. fit. edu/mmahoney/compression/rationale. html.*

Mannem, P., Prasad, R., & Joshi, A. (2010). Question generation from paragraphs at upenn: Qgstec system description. In *Proceedings of QG2010: The Third Workshop on Question Generation* (pp. 84–91).

Marcus, G. (2015). What comes after the turing test?

Marcus, M. P., Marcinkiewicz, M. A., & Santorini, B. (1993). Building a large annotated corpus of english: The penn treebank. *Computational linguistics*, 19(2), 313–330.

Markoff, J. (2011). Computer wins on 'jeopardy!': Trivial, it's not.

McKie, R. (2016). http://www.theguardian.com/technology/2016/mar/13/alphago-google-lee-se-dol.

Mitkov, R., LE AN, H., & Karamanis, N. (2006). A computer-aided environment for generating multiple-choice test items. *Natural Language Engineering*, 12(02), 177–194.

Mnih, V., Kavukcuoglu, K., Silver, D., Graves, A., Antonoglou, I., Wierstra, D., & Riedmiller, M. (2013). Playing atari with deep reinforcement learning. *arXiv preprint arXiv:1312.5602*.

Mohler, M., Bunescu, R., & Mihalcea, R. (2011). Learning to grade short answer questions using semantic similarity measures and dependency graph alignments. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1* (pp. 752–762).: Association for Computational Linguistics.

Moor, J. H. (1976). An analysis of the turing test. *Philosophical Studies*, 30(4), 249–257.

Moor, J. H. (2001). The status and future of the turing test. *Minds and Machines*, 11(1), 77–93.

Pal, S., Mondal, T., Pakray, P., Das, D., & Bandyopadhyay, S. (2010). Qgstec system description–juqgg: A rule based approach. In *Proceedings of QG2010: The Third Workshop on Ques-tion Generation* (pp.76).

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., et al. (2011). Scikit-learn: Machine learning in python. *The Journal of Machine Learning Research*, 12, 2825–2830.

Piwek, P., Prendinger, H., Hernault, H., & Ishizuka, M. (2008). Generating questions: An inclusive characterization and a dialogue-based application. In *online Proceedings of Workshop on the Question Generation Shared Task and Evaluation Challenge*: Citeseer.

Pressed, A. (2014). Computer simulating 13-year-old boy becomes first to pass turing test.

Pulman, S. G. & Sukkarieh, J. Z. (2005). Automatic short answer marking. In *Proceedings of the second workshop on Building Educational Applications Using NLP* (pp. 9–16).: Association for Computational Linguistics.

Ravichandran, D. & Hovy, E. (2002). Learning surface text patterns for a question answering system. In *Proceedings of the 40th annual meeting on association for computational linguistics* (pp. 41–47).: Association for Computational Linguistics.

Rettig, M. (1994). Prototyping for tiny fingers. *Communications of the ACM*, 37(4), 21–27.

Richardson, M., Burges, C. J., & Renshaw, E. (2013). Mctest: A challenge dataset for the open-domain machine comprehension of text. In *EMNLP*, volume 1 (pp.2̃).

Rus, V., Piwek, P., Stoyanchev, S., Wyse, B., Lintean, M., & Moldovan, C. (2011). Question generation shared task and evaluation challenge: status report. In *Proceedings of the 13th European Workshop on Natural Language Generation* (pp. 318–320).: Association for Computational Linguistics.

Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., et al. (2015). Imagenet large scale visual recognition challenge. *International Journal of Computer Vision*, 115(3), 211–252.

Russell, S., Norvig, P., & Intelligence, A. (1995). A modern approach. *Artificial Intelligence. Prentice-Hall, Egnlewood Cliffs*, 25, 27.

Searle, J. R. (1980). Minds, brains, and programs. *Behavioral and brain sciences*, 3(03), 417–424.

Shieber, S. M. (1994). Lessons from a restricted turing test. *arXiv preprint cmp-lg/9404002*.

Shieber, S. M. (2007). The turing test as interactive proof. *Noûs*, 41(4), 686–713.

Shieber, S. M. (2014). No, the turing test has not been passed.

Shieber, S. M. (2015). Principles for designing an ai competition, or why the turing test fails as an inducement prize.

Silver, D., Huang, A., Maddison, C. J., Guez, A., Sifre, L., van den Driessche, G., Schrittwieser, J., Antonoglou, I., Panneershelvam, V., Lanctot, M., et al. (2016). Mastering the game of go with deep neural networks and tree search. *Nature*, 529(7587), 484–489.

Singhal, A. (2012). Introducing the knowledge graph: things, not strings. *Official Google Blog, May*.

Sojka, P. (2010). Software framework for topic modelling with large corpora. In *In Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks*: Citeseer.

Soricut, R. & Brill, E. (2004). Automatic question answering: Beyond the factoid. In *HLT-NAACL* (pp. 57–64).

Sukhbaatar, S., Szlam, A., Weston, J., & Fergus, R. (2015). Weakly supervised memory networks. *arXiv preprint arXiv:1503.08895*.

Sukkarieh, J. Z. & Blackmore, J. (2009). c-rater: Automatic content scoring for short constructed responses. In *FLAIRS Conference*.

Taylor, W. L. (1953). Cloze procedure: a new tool for measuring readability. *Journalism and Mass Communication Quarterly*, 30(4), 415.

Tsiao, J. C.-S., Chao, D. Y., & Tong, P. P. (2007). Natural-language voice-activated personal assistant. US Patent 7,216,080.

Turing, A. M. (1950). Computing machinery and intelligence. *Mind*, 59(236), 433–460.

Varga, A. (2011). Le an ha 2010 wlv: A question generation system for the qgstec 2010 task b. In *Proceedings of QG2010: The Third Workshop on Question Generation* (pp. 80–83).

Visser, B. A., Ashton, M. C., & Vernon, P. A. (2006). g and the measurement of multiple intelligences: A response to gardner. *Intelligence*, 34(5), 507–510.

Voorhees, E. M. et al. (1999). The trec-8 question answering track report. In *Trec*, volume 99 (pp. 77–82).

Walker, M. A., Rambow, O., & Rogati, M. (2001). Spot: A trainable sentence planner. In *Proceedings of the second meeting of the North American Chapter of the Association for Computational Linguistics on Language technologies* (pp. 1–8).: Association for Computational Linguistics.

Wang, W., Hao, T., & Liu, W. (2007). Automatic question generation for learning evaluation in medicine. In *Advances in Web Based Learning–ICWL 2007* (pp. 242–251). Springer.

Weizenbaum, J. (1966). Eliza—a computer program for the study of natural language communication between man and machine. *Communications of the ACM*, 9(1), 36–45.

Weston, J., Bordes, A., Chopra, S., & Mikolov, T. (2015). Towards ai-complete question answering: A set of prerequisite toy tasks. *arXiv preprint arXiv:1502.05698*.

Winograd, T. (1971). *Procedures as a representation for data in a computer program for understanding natural language*. Technical report, DTIC Document.

Xu, Y., Goldie, A., & Seneff, S. (2009). Automatic question generation and answer judging: a q&a game for language learning. In *SLaTE* (pp. 57–60).

Yampolskiy, R. V. (2013). Turing test as a defining feature of ai-completeness. In *Artificial intelligence, evolutionary computing and metaheuristics* (pp. 3–17). Springer.

Yao, X. & Van Durme, B. (2014). Information extraction over structured data: Question answering with freebase. In *ACL (1)* (pp. 956–966).: Citeseer.

Yao, X. & Zhang, Y. (2010). Question generation with minimal recursion semantics. In *Proceedings of QG2010: The Third Workshop on Question Generation* (pp. 68–75).: Citeseer.

You, J. (2015). Beyond the turing test. *Science*, 347(6218), 116–116.

Zelle, J. M. & Mooney, R. J. (1996). Learning to parse database queries using inductive logic programming. In *Proceedings of the national conference on artificial intelligence* (pp. 1050–1055).

Ziai, R., Ott, N., & Meurers, D. (2012). Short answer assessment: Establishing links between research strands. In *Proceedings of the Seventh Workshop on Building Educational Applications Using NLP* (pp. 190–200).: Association for Computational Linguistics.