Appears in Journal of Game Development (JOGD) 3(1) pp.39-60, December 2007.

The Restaurant Game: Learning Social Behavior and Language from Thousands of Players Online

Jeff Orkin and Deb Roy
MIT Media Laboratory
20 Ames St., Cambridge, MA 02139 USA
{jorkin, dkroy}@media.mit.edu

Abstract

We envision a future in which conversational virtual agents collaborate with humans in games and training simulations. A representation of common ground for everyday scenarios is essential for these agents if they are to be effective collaborators and communicators. Effective collaborators can infer a partner's goals and predict future actions. Effective communicators can infer the meaning of utterances based on semantic context. This article introduces a computational model of common ground called a *Plan Network*, a statistical model that encodes context-sensitive expected patterns of behavior and language, with dependencies on social roles and object affordances. We describe a methodology for unsupervised learning of a Plan Network using a multiplayer video game, visualization of this network, and evaluation of the learned model with respect to human judgment of typical behavior. Specifically, we describe learning the Restaurant Plan Network from data collected from over 5,000 gameplay sessions of a *minimal investment multiplayer online* (MIMO) role-playing game called *The Restaurant Game*. Our results demonstrate a kind of social common sense for virtual agents, and have implications for automatic authoring of content in the future.

Introduction

Conversation is collaboration. The sequence of utterances, "How are you today?" "Table for one please," only makes sense because we understand the social and cultural context. This verbal exchange conjures images of a greeting between a customer and an employee of a restaurant. Taken out of context, these words appear to be a non sequitur, yet we understand them in the context of the "script" we've all learned through myriad trips to a restaurant. These scripts serve as the common ground for the collaborative activity of dialogue, which allows the two actors in the script to move jointly toward common subgoals, which may ultimately contribute to different role-dependent goals. In this case, the goal for the customer is to have a nice a meal at the restaurant, while the waitress' goal is to sell a meal at good price.

We envision a future in which conversational virtual agents collaborate with humans in games and training simulations. A representation of common ground for everyday scenarios is essential for these agents if they are to be effective collaborators and communicators. Effective collaborators can infer a partner's goals and predict future actions; effective communicators can infer the meaning of utterances based on semantic context. This article introduces a computational model of common ground called a *Plan Network*, a statistical model that encodes context-sensitive expected patterns of behavior and language, with dependencies on social roles and object affordances. We describe a methodology for unsupervised learning of a Plan Network using a multiplayer video game, visualization of this network, and evaluation of the learned model with respect to human judgment of typical behavior. Specifically, we describe learning the Restaurant

Plan Network from data collected from over 5,000 gameplay sessions of an online game called *The Restaurant Game* (Figure 1).

Figure 1 *The Restaurant Game* was developed with the *Torque* game engine [Garage Games 2006], and content from *The Sims 2* [Maxis 2004].

Schank and Abelson were the first to recognize that providing machines with representations of common ground is essential for their understanding of everyday scenarios, but in 1977, they were ahead of their time [Schank 1977]. With the technological limits of the 1970s, they could only provide common ground in the form of handcrafted scripts. It would be a simple enough task to handcraft a restaurant script where a customer sits down and says, "Bring me a steak," a waitress brings a steak, and the customer pays the bill. In reality, however, an open-ended variety of action and dialogue sequences take place in this scenario. There are limits to the range of behavior

human scripters can possibly anticipate. Handcrafted scripts are brittle in the face of unanticipated behavior, and are unlikely to cover appropriate responses for the wide range of behaviors exhibited when players are given minimal instructions to play roles in an open-ended environment. Furthermore, scripted characters have no way to detect unusual behavior. Today we can do much better.

Millions of people play video games together online. Von Ahn recognized this potential, which he leveraged with *The ESP Game* [von Ahn 2004] to collect a large image-labeling corpus. With *The Restaurant Game*, we have harnessed the power of the Internet to capture the rich behavior and language of thousands of pairs of people playing in an identical scenario. To date, *The Restaurant Game* has generated a corpus of 6,651 examples of dramatic role-play in a restaurant environment, captured from 7,504 unique players. This article describes analysis of 5,200 games collected in the first four months of the project. Each gameplay session lasts an average of 10 minutes, and consists of approximately 85 physical actions and 165 words composed in 40 lines of dialogue, produced through the interaction of two human players. In this work, we assume that any English-speaking online game player is familiar with the social conventions followed in a restaurant, and subconsciously maintains a script for expected behavior in such an establishment. Over many games, we see that a Plan Network emerges, consisting of a collection of action and utterance sequence variations that contribute to achieving common goals.

Related Work

This work draws from a range of influences, beginning with early work in cognitive psychology and chatbots, continuing through the latest innovations in computer games, language understanding, and plan learning.

[Schank 1977] introduced the idea that the human ability to understand stories and infer the missing details relies on the *scripts* we've learned from childhood. Scripts consist of roles for people and objects in the story, entry conditions, and a sequence of *scenes* that capture a chain of dependent events at an abstract level. When we read, "John went to a restaurant. He ordered chicken. He left a large tip," we can infer that he also sat down, looked at a menu, ate his chicken, and was pleased with it. These ideas were implemented by Schank's team in computer programs that relied on handwritten scripts, and thus could only make inferences for questions predetermined by their programmers. Current computer game technology allows us to simulate a restaurant at a high level-of-detail, and exploit the gameplay experiences of thousands of players to capture a wider coverage of knowledge than could be handcrafted by a team of researchers decades ago.

[Clark 1996] has theorized that language is a form of joint action. The actors in a dialogue can infer each other's meaning because they are taking part in a *joint project* with a common goal, and shared common ground. Shank's scripts are one example of common ground; knowledge of the situation we can assume everyone shares. Within the script for dining in a restaurant, there are a number of joint projects, such as seating customers, taking orders for food, and paying the check. These joint projects are collaborative activities that require participation and communication from both the customer(s) and the waitress.

[Bruner 1977] describes how young children acquire language through participating in social interaction language games. Through repeated interaction, an infant learns the social situation, which acts as scaffolding for language. Initially the language is redundant with the physical interaction. Over time, the child learns to predict actions, and is able to swap roles with the caregiver, and eventually substitute the redundant language for physical action. In a similar spirit, *The Restaurant Game* provides many examples of repeated interactions in the same social scenario. These interactions provide scaffolding for the language that players use between actions, and allow the Plan Network to learn the semantic context of utterances.

The first chatbot was a virtual therapist named ELIZA [Weizenbaum 1976]. While ELIZA's memoryless, purely associative, bag-of-words conversational system was able to fool people in the 1970s, today's audience is more sophisticated and demands more than templated, formulaic, canned responses. Recently, an experimental game *Façade* [Mateas 2005] has revisited the natural language problem, and has cast it within the context of a strong narrative. The human in *Façade* plays the role of a guest invited to a couple's apartment for cocktails, and witnesses the breakup of their marriage. The player can interact with the environment and freely type conversational text. It took two researchers five years to handcraft scripts for *Façade*, and the game takes about 15 minutes to complete. This article describes a first step toward automatically authoring a similar experience in dramatically less time by capturing human input and responses. Capturing what players actually choose to do in the environment, rather than what designers expect them to do, will provide responses to a wider range of player behaviors than a small number of script authors could possibly anticipate.

The Sims [Maxis 2000] designer Will Wright coined the term *massively single player game* to describe his upcoming creation *Spore* [*Maxis* 2008]. As opposed to a massively *multiplayer* game, like *World of Warcraft* [Blizzard 2004], where thousands of humans play online together, in a massively *single* player game, each person plays the game alone, but benefits from the participation of potentially millions of other players on the Internet. As players of *Spore* create unique creatures and environments, everything they create will be automatically uploaded to centralized servers, and shared with all other players. In this way, the game is constantly evolving on the player's desktop. Wright's approach parallels the Web2.0 movement, where users of MySpace.com and Flickr.com are empowered with tools to create shared content that benefits all other users. Luis von Ahn's *ESPGame* [von Ahn 2004] is another multiplayer game that entices people to do *Human Computation*, where people do work that is difficult for machines but easy for humans, such as labeling images. Similarly, here we are trying to harness the power of the Internet to attack the difficult problem of natural language. Like *Spore*, a game powered by a learned Plan Network would evolve and improve as people provided new conversational content by playing the game.

[Gorin 1997] demonstrated unsupervised language acquisition for AT&T's *How May I Help You* call routing system. Their system generated a database of utterances from 10,000 spoken transactions, and computed mutual information between phrases and desired actions to extract salient phrases with a high likelihood of indicating a particular type of call; for example, distinguishing between customers with billing questions, and those wanting directory assistance. [Gorniak 2005] used *Neverwinter Nights* [BioWare 2002] to demonstrate the effect of plan recognition on understanding language in the

context of a physical situation. This study collected language from pairs of humans solving a puzzle involving movement between rooms, and unlocking doors. Gorniak demonstrated that a hand-constructed probabilistic plan grammar could be used to recognize affordances and predict the next human action. The Open Mind Common Sense project [Singh 2002] uses the Internet to collect common sense facts expressed in plain English from a large number of people. This article combines these ideas, learning a Plan Network that combines action and language in a statistical model of common ground that associates relevant utterances with semantic context.

There has been previous work on learning plans including [Nilsson 1998; Shen 1994; Gil 1992; Wang 1995; Benson 1997; Zettlemoyer 2005]. What separates this work from previous efforts is that a Plan Network goes beyond learning planning rules by also learning socially acceptable sequences of plans with integrated natural language dialogue.

Designing a Role-Playing Game for High-Quality Audience Participation

We designed *The Restaurant Game* to elicit freeform dramatic role-playing from a large, diverse group of players, with enough consistency to be learnable, yet enough variety to be interesting. Our design requirements for a genre that we refer to as the *minimal investment multiplayer online* (MIMO) role-playing game stand in stark contrast to those of a typical *massively multiplayer online* (MMO) role-playing game. MMOs are typically designed to immerse players in a fantasy world where players invest many months or years customizing their characters in detail and improving them through hours upon hours of gameplay (aka *leveling up*). These characters are extensions of the players themselves, with recognizable identities and persistent reputations that reflect on the

humans controlling the avatars. Role playing in a MIMO is defined in the more literal sense of playing a role in society; a role that is expressly not an extension of self, in an experience more akin to improvisational theater than ordinary game playing. *Minimal investment* refers not only to time investment, but also to minimal investment in one particular identity with an associated reputation.

Certainly, other games exhibit minimal investment in identity, and the opportunity to play roles. For instance, players of a multiplayer online first person shooter (FPS) like *Unreal Tournament* can choose the default characters, and play roles in the game such as guarding the flag. What makes the MIMO genre unique is the combination of these characteristics with freeform dialogue, dramatic improvisation, and deeper interaction with the environment in an open-ended situation. Players of an FPS are playing roles in a collaborative effort to achieve some competitive goal, rather than dramatizing a role in society in a MIMO.

Players are placed in the familiar everyday environment of a restaurant, assigned the role of waitress or customer, and are given vague objectives of earning money or having dinner. Interaction with the world takes place through a point-and-click interface that is simple and shallow, yet consistent (Figure 2). With the exception of furniture, players can touch, pick up, examine, sit on, and eat (or at least bite) any object in the environment.

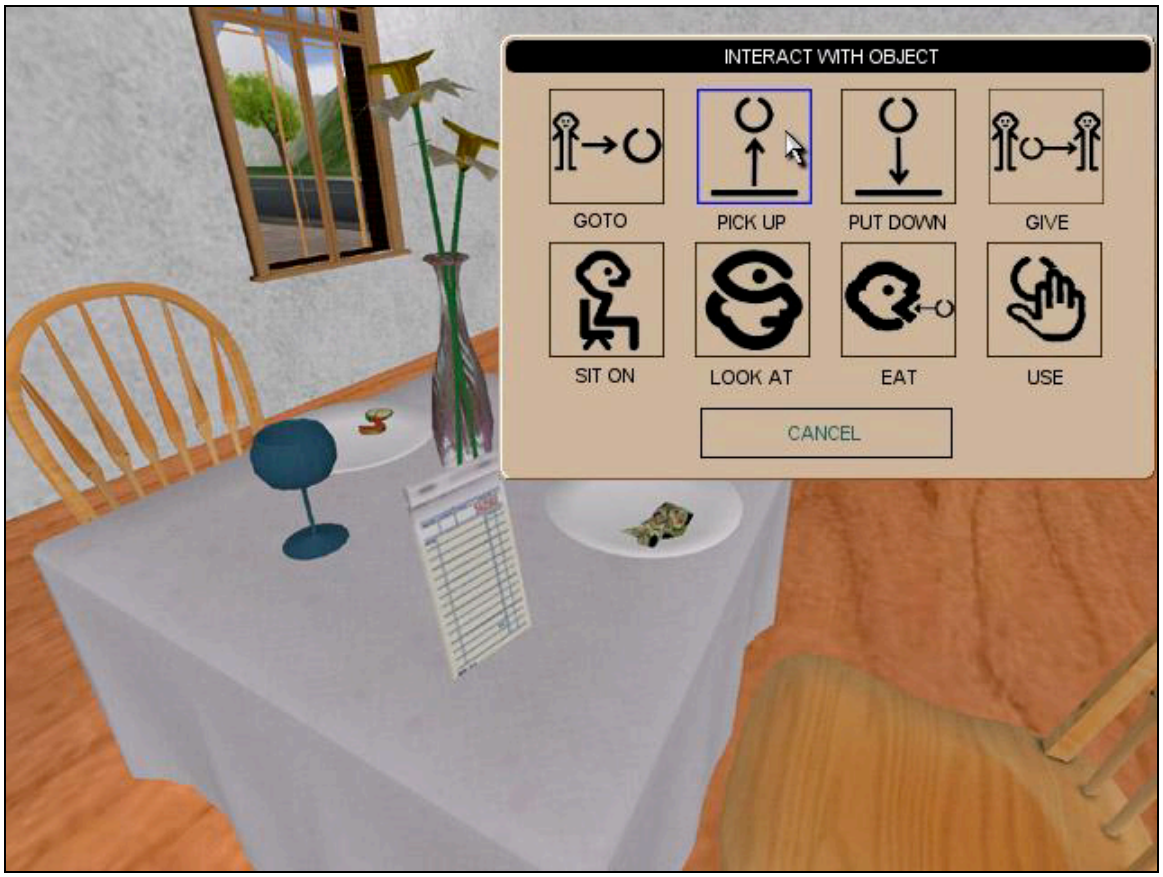

Figure 2 Interface for interacting with objects with icons inspired by *The Elephant's Memory* visual language [Housz 1994].

The open-ended design of *The Restaurant Game* subscribes to Will Wright's definition of "good game aesthetics." As Wright describes in a [Gamespot 2003] interview about designing *The Sims 2*:

> Players obviously enjoy being subversive to some degree. And so we want to provide that and let them know that we're on their side in that way […]. It's just another form of a player taking control. I think for most people, their kind of general aesthetic with games is that the more I control this experience, the better the game is […] in terms of [finding that] I can choose to go off into an interesting path, and the game will support that path. This animal we're calling subversion is really just empowering the players to not hit walls as often […]. At a fundamental level, it's kind of convergent with what I would call "good game aesthetics."

Players have, in fact, surprised us with unanticipated behavior, such as making a margarita by carrying the fruit bowl from the kitchen to the bar and using the blender. More than a few players have chosen to dramatize what happens when the customer

steals the cash register. Despite this apparent freedom, behavior tends to abide by accepted social conventions. While many players do choose to deviate from the norm, their aberrant interactions wash away statistically when compared to the larger number of examples of typical language and behavior.

Unlike avatars in an MMO, players in a MIMO are completely anonymous to one another, not even sharing usernames. While usernames themselves do not necessarily reflect gender or personality traits, they do associate players with persistent identities that develop reputations over time. MIMO games offer opportunities for rich social engagement divorced from any commitment to identity. Social roles persist, while players are free to express themselves as individuals. This detachment from identity removes any stigma from creative choices in dramatic role-play. Men can play female characters and vice versa, and playing the same character multiple times in different ways is encouraged. The post-game survey also encourages repeat playing. The survey asks players to describe who they think the other human player is in real life, in terms of age, sex, occupation, intelligence, sense of humor, honesty, consideration, eloquence, and patience. Players are presented with a cumulative personality profile based on others' impressions, which continues to evolve in subsequent play sessions.

While *The Restaurant Game* exists in the unique MIMO genre, many players bring with them habits learned from experiences in other online games. Many players try to augment the limited set of animations by typing *World of Warcraft*-style *emotes*, such as */wave, /weep, /kiss, /burp*, or */applaud*. These emotes do not currently result in any animations, but adding some of these animations might enhance the interaction, and allow the system to learn gestures, body language, and spoken language. In addition, a

number of players are familiar with the GarageGames *Torque* game engine, which powers *The Restaurant Game*, and have tried to exploit cheat commands such as *suicide* to kill and re-spawn a player, wreaking havoc on the scenario. More alarming, some of these players have attempted to run their own servers. This would be a significant problem, as the data collected by these rogue servers would be inaccessible to us for research. The lesson learned from both the emotes and the *Torque* exploits is that it is important to be mindful of the habits, experiences, and knowledge players will be bringing into the game, no matter how novel its design.

Data Collection with *The Restaurant Game*

On February 21, 2007, we started the servers and posted the project Web page at *http://theRestaurantGame.net*. The Web page includes downloadable installers for Windows and OSX, and the enticement that all players will earn Game Designer credit on a future game generated from collected data. We used several strategies to spread word of the project. We blogged on *AI-blog.net*, e-mailed colleagues at 24 software companies and 12 universities who partake in game AI research, contacted five game industry news sites and five Macintosh enthusiast sites, distributed flyers at the Game Developer's Conference (GDC) and Human Robot Interaction (HRI), listed the game on two shareware download sites, produced a YouTube gameplay video, and added Digg.com and Delicious.com badges to the project page to facilitate sharing the URL through social bookmarking. From this point, news spread organically and unpredictably, reaching unanticipated audiences including managers of the National Restaurant Association and producers of a BBC reality show about restaurant management. Other

blogs, such as Grand Text Auto, Terra Nova, and Ars Technica posted about the project. A journalist from *New Scientist* magazine wrote a short article [New Scientist 2007] about the project after picking up a flyer at HRI. PCWorld.com reviewed the game, and MSN.com later featured the review. *The Boston Globe* referenced *The Restaurant Game* in an article about AI for games [Boston Globe 2007], and National Public Radio's Ira Flatow mentioned playing the game himself in a *Science Friday* broadcast focusing on virtual worlds [NPR 2007] (Figure 3).

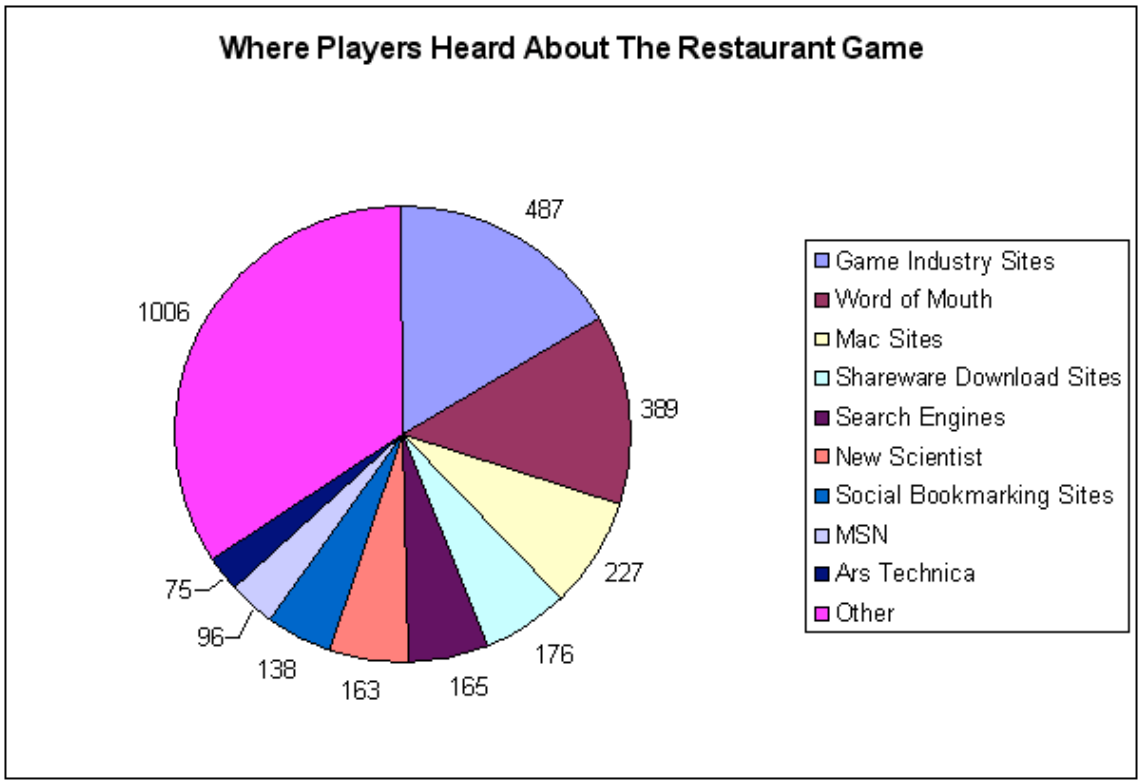

Figure 3 Where players heard about *The Restaurant Game*.

In total, 3,355 unique players completed 5,200 games between February 21 and June 25, 2007 (Figure 4). A game is considered *complete* if two players joined, both players typed chat text, and at least one player filled out a post-game survey. Based on traffic statistics from the project Web page and shareware download sites, we estimate

that about 7% of people who read about the project online actually completed a game. Players came from every continent, with the great majority from the Americas and Europe. Approximately 78% played on Windows machines, and 22% on OSX.

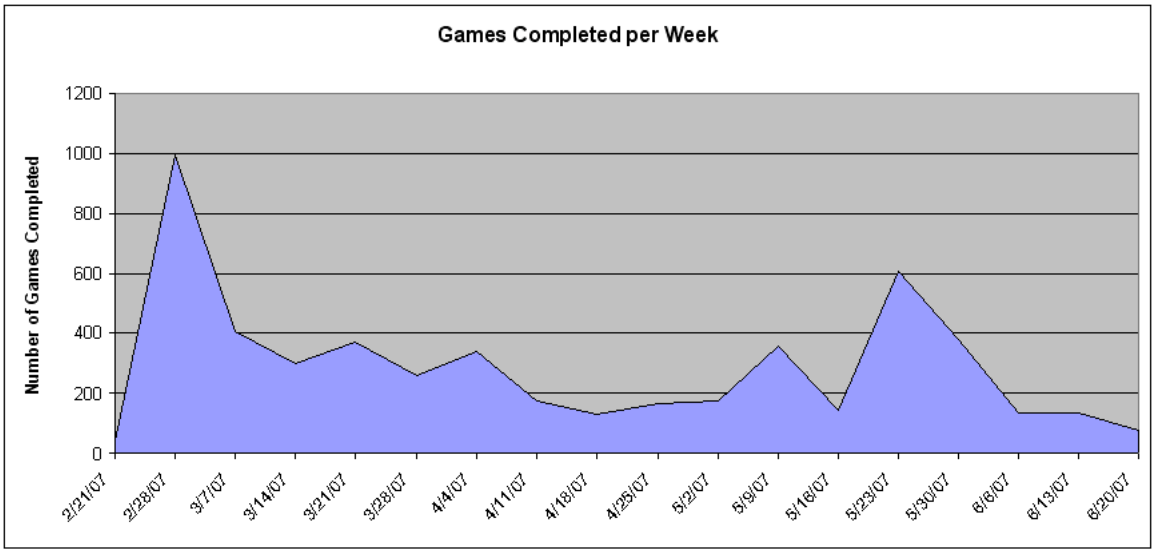

Figure 4 Games completed per week.

Visualizing Plan Networks

The Restaurant Game has captured a large quantity of data, but in its raw form, it's not of much use to a human or automated agent. Each gameplay session produces a trace of every action and utterance taken by either player, stored in a verbose text log file. It is straightforward to distill these logs into something more human-readable, but merely reading the data is not equivalent to understanding their commonalities and differences. A Plan Network provides a compact representation that incorporates everything observed in all gameplay sessions. This representation gives humans the gist of the behavior in the restaurant scenario, and provides a means of predicting future actions and recognizing when things are going "off-script" (i.e., when players behave in ways that go against norms established from commonalities of previous gameplays). Gameplay is composed

of interleaved sequences of physical actions and speech acts performed by multiple characters in various social roles. Separating the data on these dimensions provides perspectives into the data that are easier for humans to comprehend, and more powerful for planning and recognition by an agent.

Graphing Physical Behavior

You can visualize each gameplay log as a graph of physical behavior where each action is a node, and directed edges represent the ordering of observed sequences of actions. In the next section, we describe the process of generating action encodings, creating a lexicon of actions, and clustering similar actions (e.g., eating salmon and eating pie are clustered as eating *food*). Multiple logs can be combined into the same graph, where nodes are shared for actions that occur in both gameplay sessions. Figure 5 illustrates the results of merging two gameplay logs into a single graph. Each directed edge between a pair of nodes indicates that one action follows the other temporally. Nodes and edges are either red or blue, depending on the gameplay session in which they were observed. Nodes and edges appearing in both sessions are bold purple. Nodes are labeled with the role, action, and interaction object, followed by a varying number of contextual variables in parentheses. For example, a node labeled WAITRESS_PICKUP_Food(Counter, STAND) indicates that a waitress picked up food from the kitchen counter while standing.

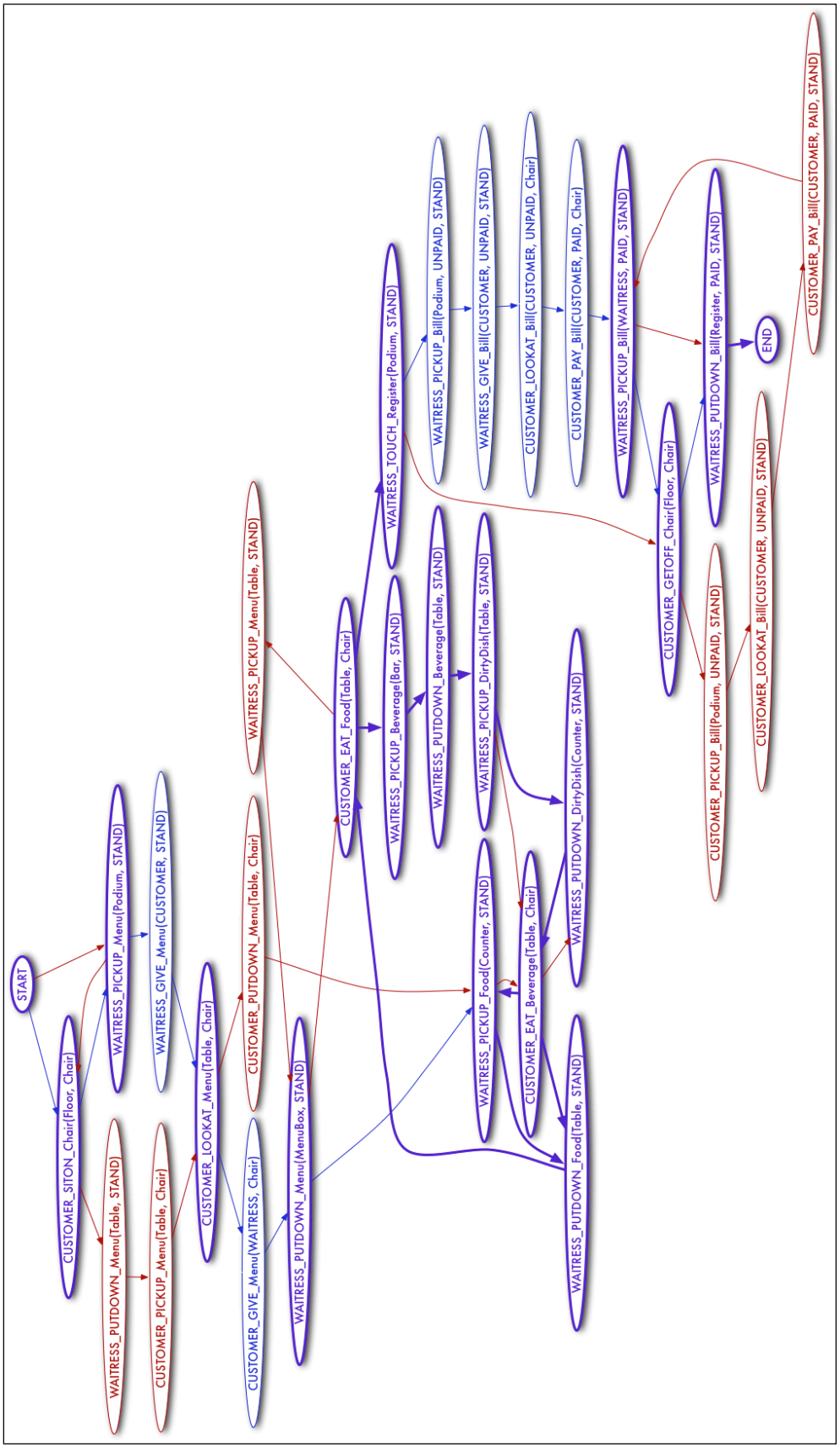

Figure 5 Graph visualization merging two games.

This process can be repeated for every captured gameplay session to generate a comprehensive graph of the action possibility space for the restaurant scenario (Figure 6). The variety of actions and orderings results in a dense mesh of edges between nodes, illustrating the huge amount of variation in the way pairs of people choose to play the game. The graph is so complex, in fact, that it is difficult to assess whether it is an accurate depiction of what people normally do in a restaurant. The graph includes expected restaurant behavior such as waitresses giving drinks to the customers, and depositing bills in the cash registers, and atypical behavior like customers sitting on tables and picking up flower vases.

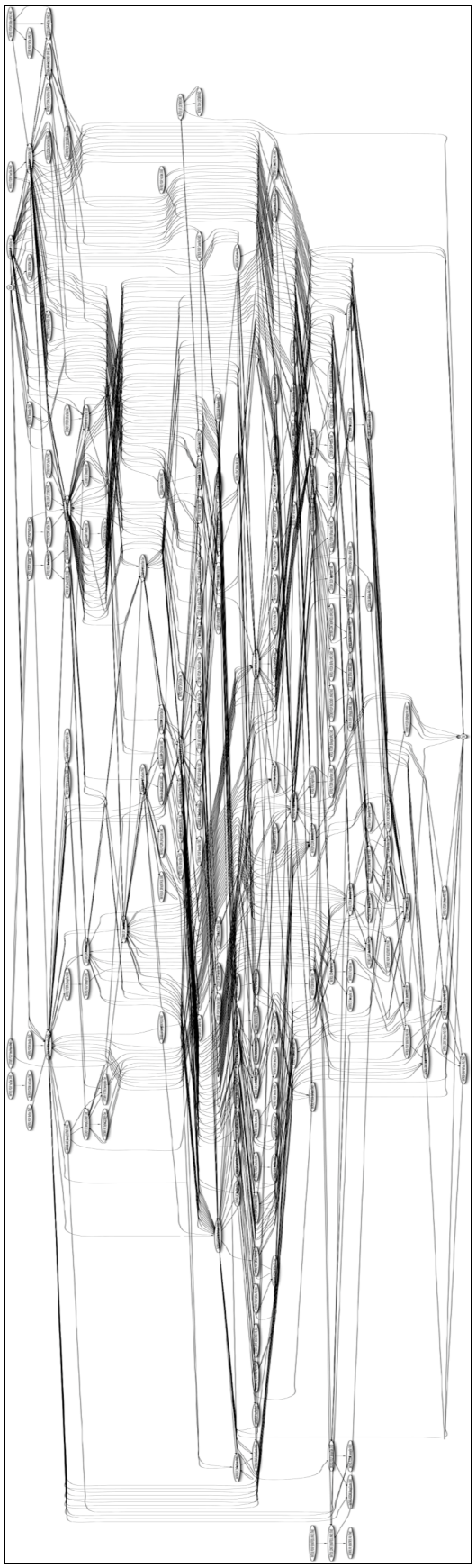

Figure 6 Graph visualization merging 5,000 games.

These visualizations can be greatly improved by making use of statistics captured in the Plan Network and by leveraging social roles. Two thresholds in our system allow us to filter atypical behavior out of our visualizations: α and β. The Quantitative Evaluation section describes the process of estimating the likelihood of each gameplay session, and selecting an α threshold. Games with a likelihood estimated to be less than the α threshold are filtered out of the visualization entirely. We set α=0.04, which filtered out about 35% of the games. The remaining 65% of our data was considered representative of typical behavior, and worthy of visualizing. With this remaining data, we catalogued every observed transition between a pair of actions, and computed the percentage of games that contain each transition. Visualizations contain nodes and edges associated with action transitions observed in a percentage of games above the β threshold. We set β=0.065, resulting in only visualizing transitions that appear in more than 6.5% of the typical games that remain after applying the α threshold. This β setting filters out about 98% of the observed action transitions from the visualization. Interleaving the actions of the two actors produces a lot of noise, because much of the time, these two are acting in parallel, and one's next action does not depend on the other. Figure 7 graphs behavior for a single social role, the waitress, with statistically unlikely behavior filtered out.

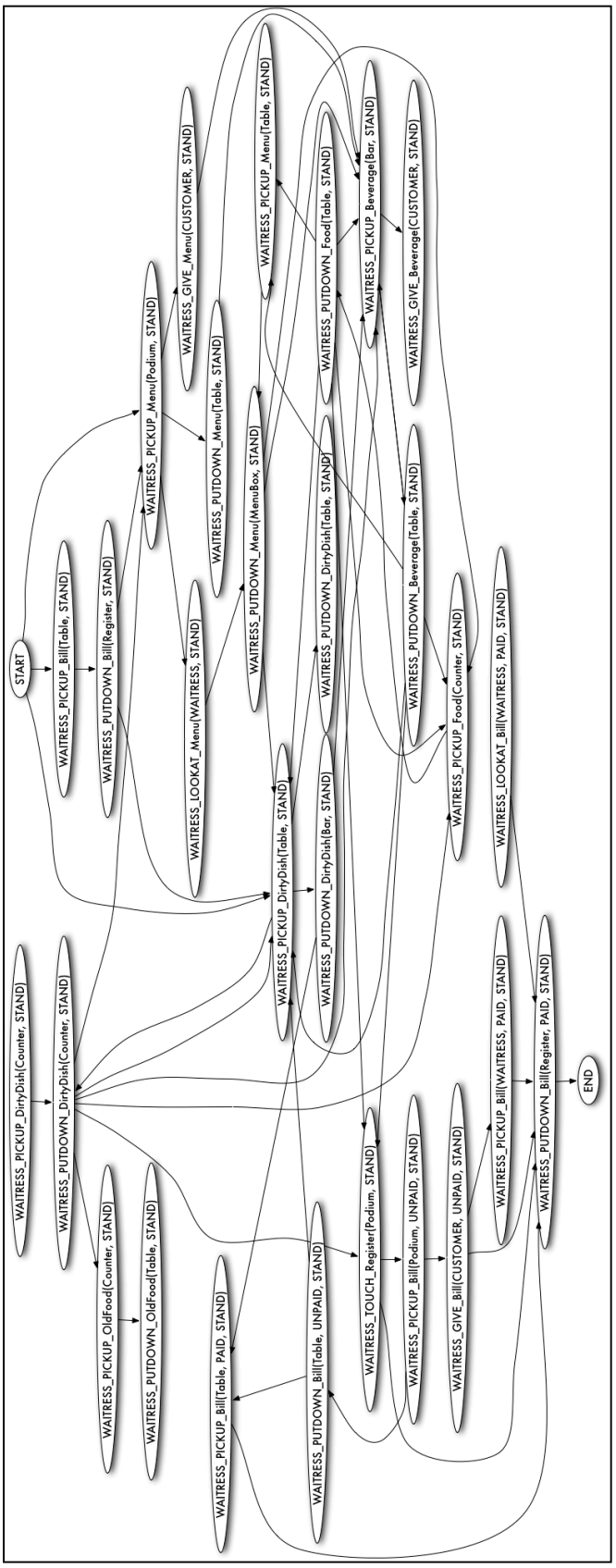

Figure 7 Filtered graph of only waitress behavior from 5,000 games.

This graph of filtered waitress behavior is much simpler than the previous graph, yet still gives the gist of behavior learned from 5,000 games. Beyond visualization, filtering gives the system a sense of typical behavior, and the separate models of social roles could lead to a kind of "mental model" of others for an automated agent, useful for inferring goals of a collaboration partner. The graph allows verification that the learned behavior is what one would expect from a waitress in a restaurant. Prior to the arrival of a customer, waitresses clean off tables and collect paid bills. Once a customer arrives and sits down, the waitress gives him a menu. After serving a customer, the waitress might serve him more food, or create a bill at the cash register (by touching the register). Finally, the waitress puts the bill on the table for the customer and picks it up for deposit once it is paid.

Browsing Conversations

Graphing models of physical behavior give humans the gist of the behavior captured in our data. This section explores how to get a similar view of linguistic data. Taking inspiration from Bruner's theory, the graphical model of physical behavior can provide the scaffolding that lends meaning to utterances. Between any two physical action nodes, all conversations uttered between players can be clustered. Figure 8 is a screenshot of the conversation browser application that runs alongside *The Restaurant Game*.

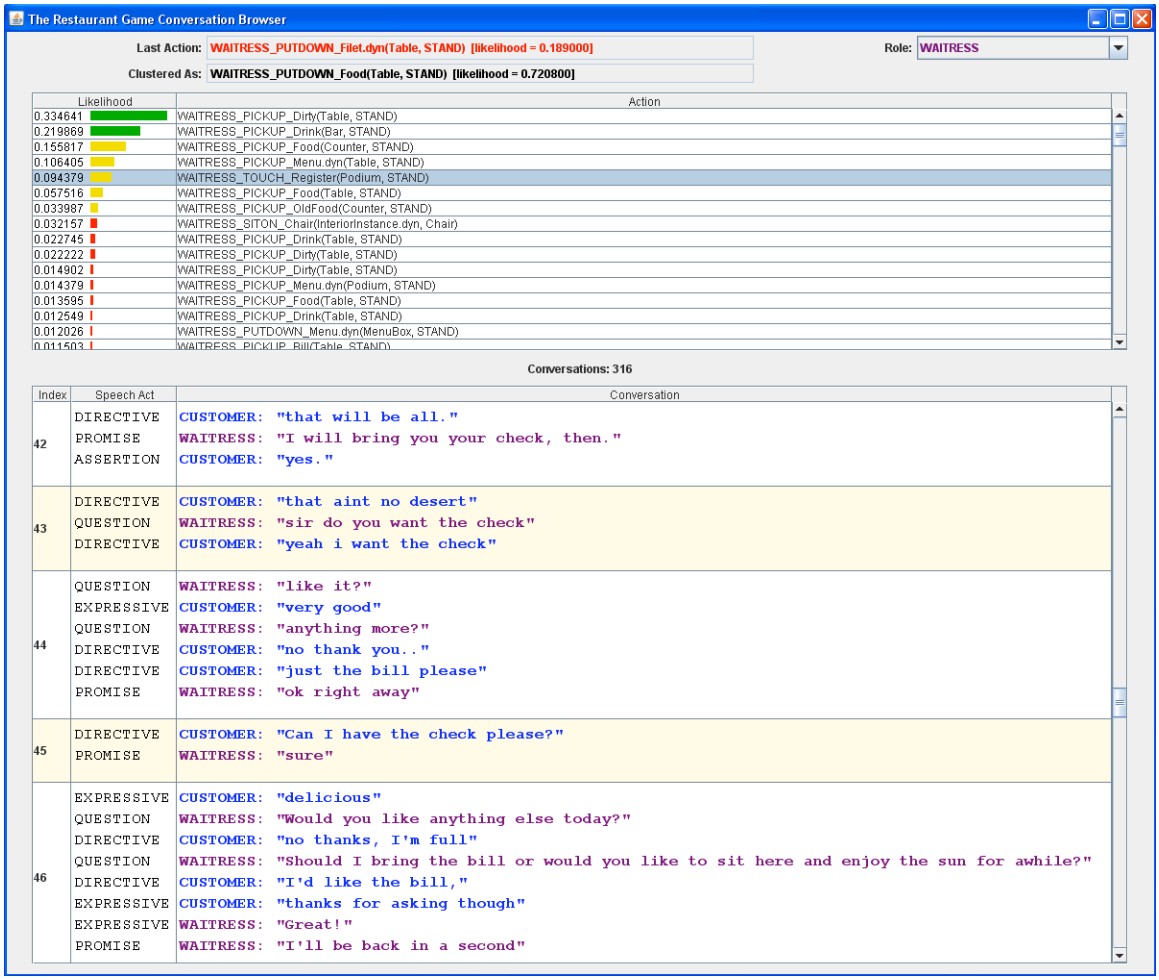

Figure 8 Browsing conversations after putting down food, captured from 5,000 games.

In this screenshot, the browser recognizes that it is highly probable for a waitress to put food on a table, and provides a sorted list of likely subsequent actions. She might clean up some dirty dishes, pick up a drink from the bar, bring more food, or ring up the customer at the register. If the user selects touching the cash register (to create a bill) as the next action, the browser displays 316 unique examples of conversations that precede the waitress operating the register. These conversations typically include directives from the customer referring to a check or bill.

Equally as interesting as the conversations clustered within the likely branches of the physical action model are those that exist within unlikely branches. The following

example bolsters the claim that capturing behavior from human players produces a more robust model than hand-authored scripts. There is a fruit bowl in the back of the kitchen sitting on the dishwasher. This is a purely decorative prop that would be noninteractive in most games, or at best physically simulated to fall down when pushed. The system recognizes that the waitress picking up the fruit bowl is highly unlikely; however, in the event that she *does* pick it up, the system predicts that she is most likely going to place it on a table (as she did 63% of the time that she picked up the fruit), and displays 37 examples of conversations that occur when she does. In one game, the customer compliments the bananas. In others, the waitress offers a complimentary bowl of fruit. The Plan Network captures dialogue that human scripters are unlikely to include, because the Plan Network captures what people really do, rather than scripting in anticipation of what they are expected to do.

Constructing Plan Networks

The automatic, bottom-up process of constructing a Plan Network consists of first compiling a lexicon of physical actions and a lexicon of words, next clustering physical actions to simplify the network, and finally computing frequencies of actions, words, and transitions between actions and between words.

Action and Language Lexicons

An action is represented in a logical format similar to a STRIPS operator, as described in [Fikes 1971]. Each action is defined by a set of preconditions and effects, where preconditions are world state variable criteria that must be met to activate this action, and effects are assignments to world state variables upon completion. In addition, every

action has two required parameters: a social role requirement, and an interaction object. Here is an example of an action representing a customer picking up a salad (with one bite taken) from a table while sitting in a chair:

```
ACTION: PICKUP
 REQUIREMENTS:
 ROLE = CUSTOMER
 OBJECT = Salad
 PRECONDITIONS:
 ACTOR::SITTINGON = Chair
 OBJECT::ATTACHEDTO = NULL
 OBJECT::ON = Table
 OBJECT::SHAPE = Bite1
 EFFECTS:
 OBJECT::ATTACHEDTO = CUSTOMER
 OBJECT::ON = NULL
```

Action encodings with this representation are generated automatically as the gameplay log files are processed, and preconditions are assigned based on the current state of the world at the time the action's execution was observed. The scope of the world state considered for preconditions is limited to the acting agent and the object being acted upon. All state variables associated with the action's actor and interaction object are included with each action. This agent-centric world state representation is similar to that described in [Orkin 2005]. Each unique action encoding is added to the *action lexicon*.

Many of the objects in the world of the restaurant are functionally similar. If these objects can be clustered, the actions that refer to these objects can also be clustered, greatly reducing the size of our action lexicon. Objects are clustered automatically by counting the frequencies of role-specific actions on objects, and using these frequencies to calculate interaction probabilities. An interaction includes an actor, an interaction object, and additional objects that provide context. The kitchen counter provides context for a waitress picking up food, and the bar provides context for the waitress picking up

drinks. Interactions with probabilities above an empirically determined threshold are considered an object's *affordances*, and objects with identical lists of affordances are clustered. For instance, chairs are clustered with stools because customers sit on them, and salmon is clustered with steak because customers eat them and waitresses pick them up from the counter. Some objects do not cluster well due to lack of interaction data, such as pots and pans, blenders and food processors. This is not a significant problem since the lack of data indicates that these items are less significant to the scenario we are trying to learn. The benefit of clustering is that the number actions in our action lexicon can now be greatly reduced by referencing clusters rather than individual items. After 5,000 games, the action lexicon has 7,086 clustered actions, and 11,206 total actions. For more implementation details on the process of clustering and building the action lexicon, see [Orkin 2007].

Building a language lexicon is more straightforward. As new words are encountered while processing gameplay logs, these words are added to the language lexicon. Games contain an average of 40 speech acts, each consisting of an average of four words. After 5,000 games, the language lexicon has 23,691 words. The maximum language lexicon size is essentially unlimited due to slang, typos, abbreviations, tenses, and multiple languages. However, the growth is less than linear due to some limitation on the range of words people typically use in a restaurant scenario (Figure 9).

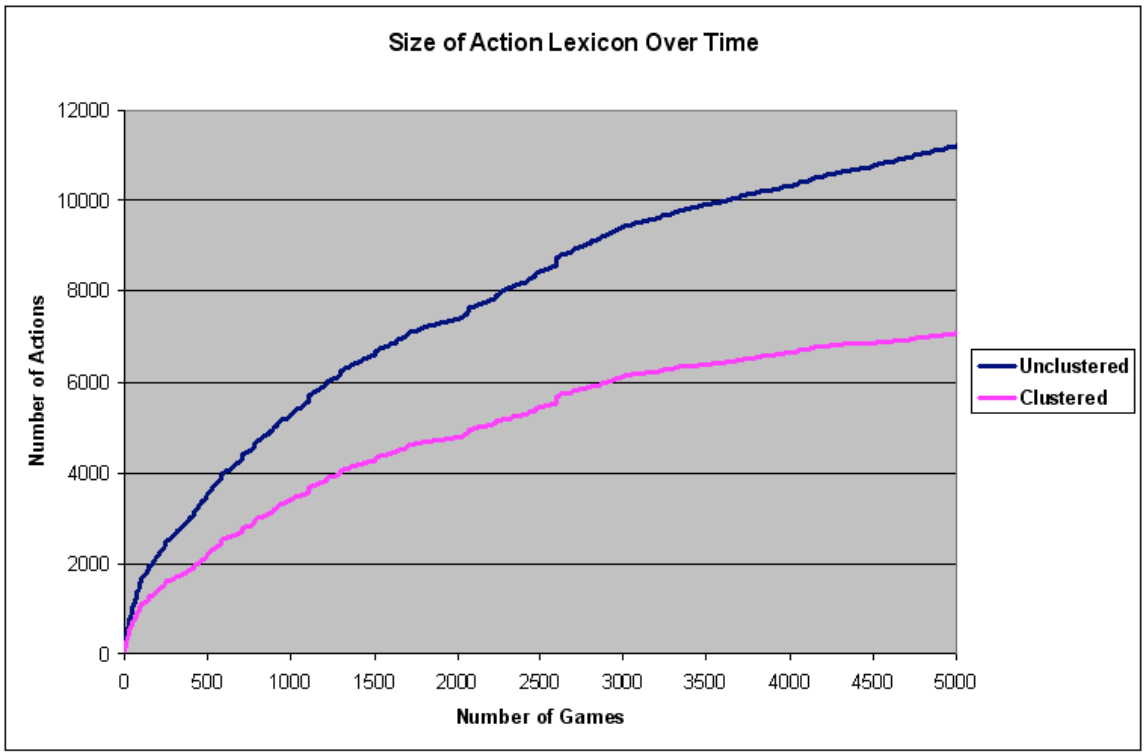

Figure 9 Growth of action lexicon, clustered and unclustered, over 5,000 games.

N-gram Models of Language and Behavior

N-grams are a commonly used statistical modeling technique used in the field of Natural Language Processing. Applying this technique to both the language lexicon and action lexicon gives a means of estimating the likelihood of new gameplay sessions, and quantitatively evaluating how well the learned model of restaurant behavior and language correlates with human judgments.

An n-gram model estimates the likelihood of a sequence of words of length N by computing the conditional probability of observing the Nth word given the previous N-1 words. Unigrams estimate the likelihood of atomic words; bigrams estimate the likelihood of pairs of words; trigram estimate triplets, and so on. The likelihood of an entire sentence can be estimated by iterating over every sequence of N words, and

multiplying the n-gram probabilities together. For example, estimating the likelihood of the sentence "The dog laughs." with trigrams looks like this:

$$P(\text{the, dog, laughs}) = P(\text{the} \mid \text{START}) \times P(\text{dog} \mid \text{START, the}) \times P(\text{laughs} \mid \text{the, dog}) \times P(\text{STOP} \mid \text{dog, laughs})$$

The trigram $P(\text{laughs} \mid \text{the, dog})$ is the probability of seeing "laughs" given that we have already seen "the dog." Enclosing each sentence in START and STOP markers captures the probability of starting or ending a sentence with a particular n-gram.

If all of the actions in an entire gameplay session are treated as one sentence, an n-gram model can be used to estimate the likelihood of this sequence. Similarly, all of the words in all of the speech acts in one gameplay session can be concatenated into one long sentence, to estimate the likelihood of the sequence of words in one game. This provides two "lenses" through which new gameplay sessions can be examined to assess their typicality. If the likelihoods assigned by the action and language models correlate well with ratings of typicality by humans, this quantitatively shows that the learned Plan Network is a useful representation of how humans think about the restaurant scenario.

The probability of an individual action or word is computed by dividing the count of games in which the action or word appears by the total number of games. For example, if we observe customers sitting on chairs in 4,000 out of 5,000 games, the unigram probability $P($`CUSTOMER_SIT_Chair`$) = 4{,}000 / 5{,}000 = 0.8$. We compute the probability of an n-gram of length greater than one by dividing the count of the complete n-gram by the count of the n-gram one shorter. If we observe the trigram sequence `CUSTOMER_SIT_Chair -> CUSTOMER_LOOK_Menu -> CUSTOMER_EAT_food` in 1,000 games, and the bigram sequence

`CUSTOMER_SIT_Chair -> CUSTOMER_LOOK_Menu` in 3,000 games, then *P*(`CUSTOMER_EAT_food`|`CUSTOMER_SIT_Chair, CUSTOMER_LOOK_Menu`) = 1,000 / 3,000 = 0.33. In this example, the sequence `CUSTOMER_SIT_Chair -> CUSTOMER_LOOK_Menu` is followed by `CUSTOMER_EAT_food` in 1,000 games, and is followed by some other action in the remaining 2,000 games that contain the bigram sequence. Note that in these examples, associated state variables are omitted from the action descriptions for brevity.

The probability calculations need to incorporate discounting and smoothing techniques to counter the effects of data sparsity. To estimate the likelihood of previously unseen action or word sequences or atoms, true counts are discounted in some way, and the missing probability mass is distributed among previously unseen n-grams. We implemented Katz Back-Off smoothing as described in [Jurafsky 2000]. The Katz method computes a discounted maximum likelihood estimate for bigrams that do exist in the corpus, and backs-off to an estimate defined in terms of unigram maximum likelihoods for previously unseen bigrams. Similarly, the Katz method computes a discounted maximum likelihood estimate for existing trigrams, and backs-off to an estimate defined in terms of Katz bigrams for previously unseen trigrams. This back-off technique continues up to the desired length of n-grams, each backing-off to an estimate defined in terms of Katz n-grams one word smaller.

Quantitative Evaluation

We evaluated the system quantitatively by computing likelihood estimates for gameplay sessions based on a combination of physical actions and words, and correlating these

scores with human ratings of the same sessions. The intuition is that if the system has learned a high-quality Plan Network for the restaurant scenario, it should exhibit a human-like capability to judge typical and atypical behavior in the restaurant environment. The capability to judge typicality is a powerful asset of a Plan Network that enables users of the system to automatically detect off-script behavior; irregularities that would be undetectable with a handcrafted system. Behavior and language estimated to be highly likely by the Plan Network represents the behavior and language that can be expected from an agent driven by the learned model. Behavior and language estimated to be unlikely provides a red flag, indicating that some action may be required on the part of an agent to steer the scenario back toward typical behavior or take other appropriate action.

From a pool of 5,200 games, we randomly selected 400 games to set aside for validation and testing. We divided the 400 games into a 100-game validation set for use in tuning the system, and a 300-game test set. The remaining 4,800 games made up the training set. None of the 400 validation or test games were used to train the system. We tuned the system to determine the best configuration prior to testing. We first describe our test results, and then describe the preceding tuning process.

Ten people rated gameplay sessions from the test set on a 1-to-7 scale of typicality, where 7 represents a perfect example of what one would expect to witness in a restaurant, and 1 indicates that the behavior was not at all representative of what people do in a restaurant. Each person rated a unique set of 30 log files (formatted for human readability), for a total of 300 games. These judges also rated a common set of 10 games used to measure inter-rater reliability. Strong inter-rater agreement was found, measured

at 0.86 with Light's weighted Kappa for N raters, where 1.0 indicates perfect agreement. We trained a Plan Network based on our 4,800-game training set, and measured the correlation between the likelihood estimated by the system for the 300 test games, and the human ratings for these games. The result was a Pearson's R = 0.576 correlation coefficient, which is strongly significant for 300 games. With 300 games, a correlation of 0.147 or greater is significant at the 0.01 level ($p<.01$).

It is notable that the system is at a disadvantage because it is only trained on behavior observed in the virtual restaurant, while humans are asked to rate the typicality as compared to real-world restaurant behavior. Interestingly, we still find a strong correlation between the system's likelihood estimation and the human ratings of typicality, suggesting that when humans are placed in a familiar environment, they tend to bring their social conventions with them from the real world into the virtual.

Based on the scatter plot in Figure 10, we determined that an α threshold of 0.04 does a good job of filtering out the majority of atypical examples. This is the same α mentioned previously in the Visualization section. Table 1 details the system's performance on the binary classification task of identifying typical and atypical gameplay sessions in the test set based on setting α=0.04. Humans rated 132 games as typical (99 + 33 = 132), 117 games as atypical (85 + 32 = 117), and 51 games as neutral (not included in Table 1). The Plan Network correctly identifies 75% of the games that humans rated as typical (99 / 132 = 0.75), and filters out 73% of the games that humans found to be atypical (85 / 117 = 0.73).

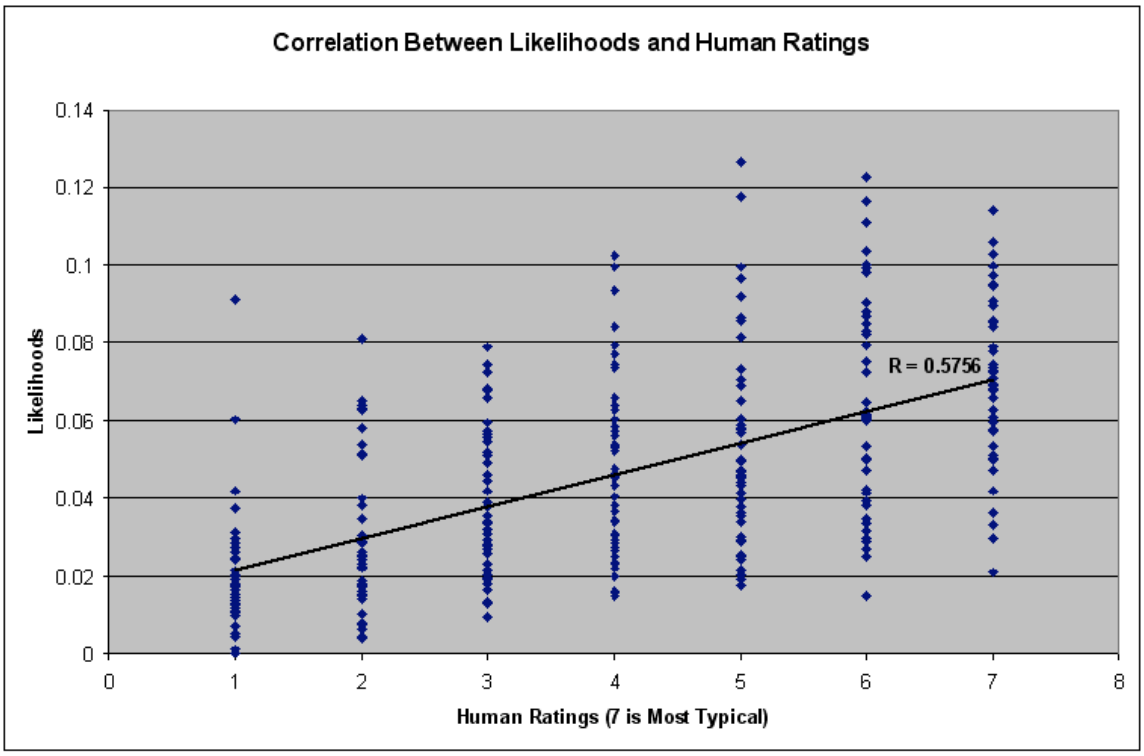

Figure 10 Scatter plot of correlation between likelihoods and human ratings.

Table 1 Binary Classification Matrix Comparing Estimated Likelihoods to Human Ratings

	Rating > 4 (Typical)	Rating < 4 (Atypical)
Likelihood > 0.04 (Typical)	99 True Positives	32 False Positives
Likelihood <= 0.04 (Atypical)	33 False Alarms	85 True Negatives

There are a number of ways to combine models of behavior and language from the restaurant scenario to estimate likelihood. Prior to testing the system, we found the best combination through experimental tuning on the validation set of 100 games, and the test results reported in Table 1 were based on the tuned system. We measured the correlation between human ratings and n-gram models of language and behavior from unigrams to 6-grams, and found the best correlation with a bigram model of language, and a 4-gram model of behavior. For both behavior and language, an estimate based on averaging likelihoods from separate waitress and customer models achieved a better correlation with human ratings than likelihood estimates based on an interleaved model.

This result is related to the noise observed when visualizing the interleaved model. The best correlation overall was achieved by interpolating a likelihood estimate between the averaged language and behavior models, with a distribution of 25% physical actions and 75% words, challenging the old adage that actions speak louder than words. This is an interesting counter-intuitive result, given that the action model consistently outperforms the language model before interpolating the two (Figure 11).

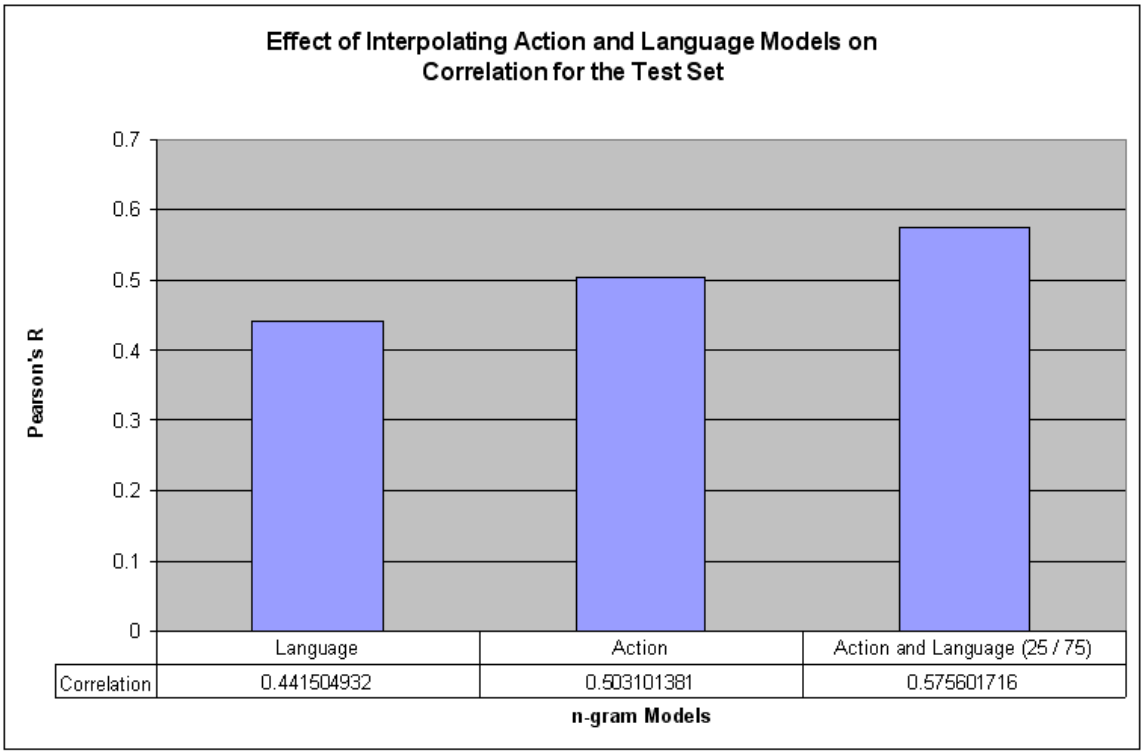

Figure 11 Effect of interpolating action and language models on correlation for the test set.

System Successes and Failures

Figure 12 illustrates the Plan Network detecting off-script behavior in a game rated as a 3 by a human, and below 0.04 by the system. The customer stands up after looking at the menu, and then sits down again. Even more alarming, the customer picks up the flower from his table, and then grabs the cash register before exiting the restaurant—without

paying his bill! The typical behavior follows pre-existing paths through the customer's model, highlighted with blue nodes. The off-script actions and decisions are illustrated with red nodes and edges inserted manually into the automatically generated visualization.

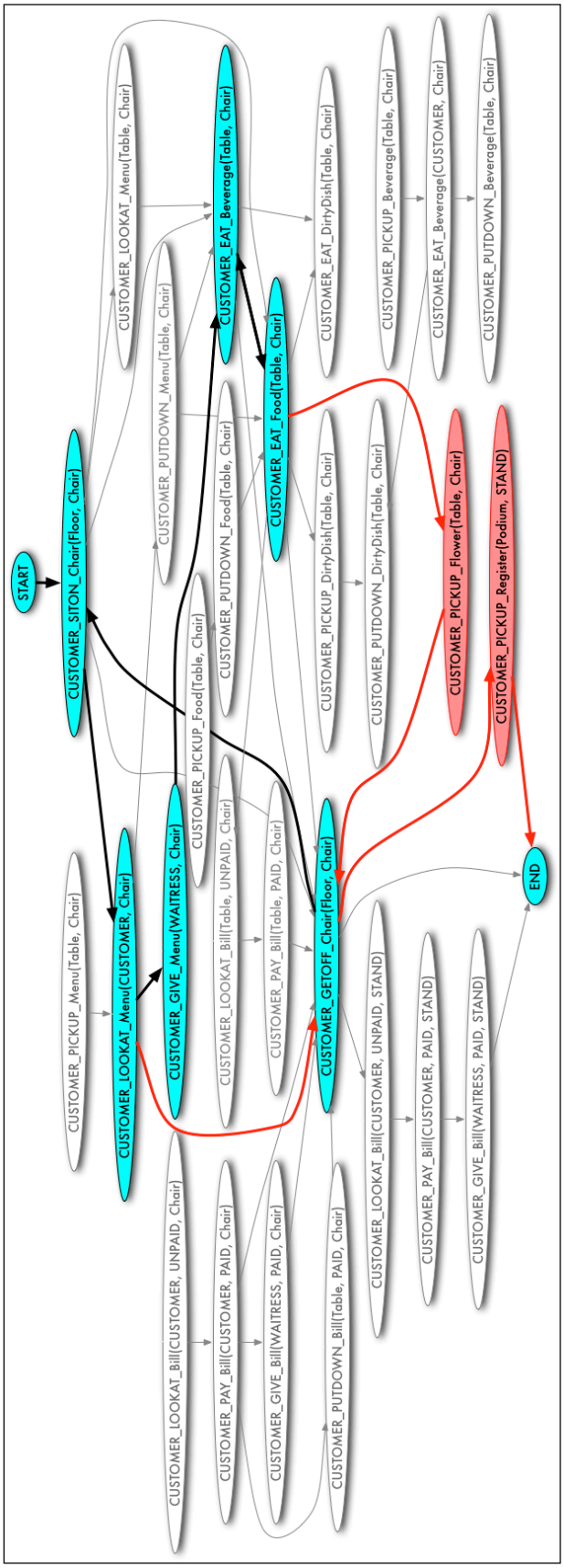

Figure 12 Graph of customer behavior in a game rated atypical by humans and the system.

The graph only represents the physical behavior. It is possible to find games in which the physical behavior is typical, but the language is not, and the system can successfully identify these as well. For example, in one game, we observe the waitress and customer engaging in ordinary behavior such as eating food and paying bills, but the dialogue between them reveals that the customer is trying to leave without paying, and trying to take the waitress home with him. Here the human rating is 2, and the system estimates a likelihood of below 0.03. The typicality of physical behavior is estimated as highly likely with a score over 0.09, but the unusual language forces the combined likelihood below the threshold of typical behavior.

Relying on statistics and Markovian assumptions will not always lead to successful detection of typical behavior. There are a number of ways to confuse the system. Sometimes, players engage in conversation about topics outside of the scope of the scenario, introducing new vocabulary into an otherwise typical game. The unrecognized words lead the system to assign a low likelihood to a game that humans rate as typical. Other times, players repeat typical restaurant language and behavior an atypical number of times. A human assigns a rating of 2 to a game where the waitress and customer cooperatively fill the restaurant with pie and beer, but the system's failure to assign a low likelihood highlights the limitations of using an n-gram model that is unable to encode long-term dependencies in sequences. Even if we were to use higher order n-gram sequences (i.e., larger values of n), ordering several dishes and beverages is not out of the ordinary, but ordering the *same* item multiple times is. This is an example where the loss of precision due to clustering leads to a false positive.

Atypically short or truncated games composed of typical language and behavior can also be problematic. Humans evaluate games based on what did and what did *not* happen, and can recognize a game as strange when the customer refuses service and exits the restaurant without ordering, eating, or paying a bill. The system needs some representation of the higher-level structure of a game to recognize that it is out of the ordinary for a customer to stand up and exit without having a meal. The current implementation only has access to a local representation of the expected sequence of events; it cannot identify gross absences.

Contributions and Future Work

The results presented in this article demonstrate that it is possible to not only collect a large quantity of data from a multiplayer video game, but also data of high quality that accurately reflects typical human behavior and language. We described a methodology to learn a Plan Network from this data that can be visualized as a graph, and interactively browsed to view conversations clustered by context. We applied a quantitative measure to a random selection of 300 games, positively correlating the Plan Network's estimate of likelihood with ratings from 10 human judges.

The Restaurant Game is a noncommercial MIMO role-playing game quickly constructed for data collection purposes, and is hardly a game at all. Yet, this restaurant simulation provided enough entertainment to capture behavior from well over 7,000 people in just six months. With hundreds of thousands of people online in *Second Life*, and millions playing *The Sims* and *World of Warcraft*, the potential for using games to teach AI characters about humans is enormous.

The data analysis performed so far only scratches the surface of a rich research area. The system failures in the previous section illustrate the need for additional representations of learned behavior that capture the large-scale structure of the narrative, and are capable of detecting atypical omissions, such as exiting the restaurant without paying the bill.

This article began with the statement that conversation is a collaboration, yet the best likelihood estimations are achieved by learning separate models for the two social roles. Ultimately, an additional unified model of the interaction dynamics between the customer and waitress is desirable, because an agent whose behavior is driven by the Plan Network will need to understand when to expect the other actor to take a turn. The nodes of the Plan Network already contain the knowledge required to build the unified model, in the form of action preconditions. For example, the customer's behavior model includes an edge from looking at the menu to eating food. In between, a waitress must put food on the table for the customer to eat. The preconditions of eating food require food to be on the table, providing the required knowledge that interaction between the waitress and customer is required between looking at the menu and eating food. Implementation of the unified model of interaction dynamics remains for future work. Ideally, the Plan Network will also scale to handle more than two simultaneous actors.

Despite the issues raised previously, the Plan Network in its current form is already useful. Plan Networks allow humans to visualize norms that emerge from thousands of people interacting in the same environment, and may expose unexpected behavior or user interface confusion. The capability to recognize atypical behavior and language enhances visualizations, and can filter data used to teach an agent appropriate

behavior, and to recognize off-script behavior in real time. Perhaps the chef starts a small fire in the kitchen to distract a player who was detected taking the scenario in an atypical direction with erratic behavior or out of vocabulary language. Course correction like this may improve games, but would be especially useful in training simulations; one of the many potential application areas for socially aware, conversational role-playing agents. In the future, conversational agents powered by Plan Networks may provide new experiences in gameplay and training, allow new ways to practice foreign languages, and act as digital extras in animated films and machinima.

Of course, there is more work to be done before Plan Networks can drive agents in interactive applications. A future goal for this work is to automate conversational characters in a new single-player restaurant game, fulfilling the promise made to participants on the project Web page.

References

[Benson 1997] Benson, S., "Learning Action Models for Reactive Autonomous Agents." Ph.D. dissertation, Stanford University Computer Science Department.

[BioWare 2002] BioWare. *Neverwinter Nights*. Atari.

[Blizzard 2004] Blizzard Entertainment. *World of Warcraft*. Vivendi Universal Games Inc.

[Boston Globe 2007] "Game Designers Test the Limits of Artificial Intelligence." *The Boston Globe*, June 17, 2007.

[Bruner 1977] Bruner, J., "Early Social Interaction and Language Acquisition." In H.R. Schaffer (Ed.), *Studies in Mother-Infant Interaction*. New York: Academic.

[Clark 1996] Clark, H. *Using Language*. Cambridge University Press.

[Fikes 1971] Fikes, R.E. and Nilsson, N.J. (1971). STRIPS: A new approach to the application of theorem proving to problem solving. Artificial Intelligence, 2(3–4), 189–208.

[Gamespot 2003] Gamespot, "Q&A: Mapping Will Wright." *www.gamespot.com/pc/strategy/thesims2/news_6085945.html*.

[Garage Games 2006] Garage Games, *Torque Game Engine v1.5 (Windows and Universal Binary Mac OSX)*. *www.garagegames.com*.

[Gil 1992] Gil, Y. "Acquiring Domain Knowledge for Planning by Experimentation." Ph.D. dissertation, School of Computer Science, Carnegie-Mellon University.

[Gorin 1997] Gorin, A., Riccardi, G., and Wright, J., "How may I help you?" *Speech Communication*, Volume 23. Elsevier Science.

[Gorniak 2005] Gorniak, P. and Roy, D., "Probabilistic Grounding of Situated Speech using Plan Recognition and Reference Resolution." *Seventh International Conference on Multimodal Interfaces* (ICMI 2005).

[Housz 1994] Housz, T. I., "The Elephant's Memory: An Interactive Visual Language." *www.khm.de/~timot/PageElephant.html*.

[Jurafsky 2000] Jurafsky, D. and Martin, J. *Speech and Language Processing*. Prentice Hall.

[Mateas 2005] Mateas, M., and Stern, A., "Procedural Authorship: A Case Study of the Interactive Drama Façade." *Digital Arts and Culture* (DAC 2005).

[Maxis 2000] Maxis. *The Sims*. Maxis/Electronic Arts Inc.

[Maxis 2004] Maxis. *The Sims 2*. Maxis/Electronic Arts Inc.

[Maxis 2008] Maxis. *Spore*. Maxis/Electronic Arts Inc.

[New Scientist 2007] "Virtual Dining Provides Tips for Cyber Characters." *New Scientist*, March 24, 2007. Reed Business Information Ltd.

[Nilsson 1998] Nilsson, N. *Artificial Intelligence: A New Synthesis*. Morgan Kaufmann Publishers.

[NPR 2007] "Go Get a (Virtual) Life." *Talk of the Nation: Science Friday*, August 31, 2007. National Public Radio.

[Orkin 2005] Orkin, J., "Agent Architecture Considerations for Real-Time Planning in Games." *Artificial Intelligence & Interactive Digital Entertainment* (AIIDE 2005).

[Orkin 2007] Orkin, J., "Learning Plan Networks in Conversational Video Games." M.Sc. in Media Arts and Sciences Thesis, MIT Media Laboratory.

[Schank 1997] Schank, R., and Abelson, R. *Scripts, Plans, Goals, and Understanding: An Inquiry into Human Knowledge Structures*. Lawrence Erlbaum Associates.

[Singh 2002] Singh, P., Lin, T., Mueller, E., Lim, G., Perkins, T., and Li Zhu, W., "Open Mind Common Sense: Knowledge acquisition from the general public**."** *Proceedings of the First International Conference on Ontologies, Databases, and Applications of Semantics for Large Scale Information Systems*. Irvine, CA.

[Shen 1994] Shen, W.-M. *Autonomous Learning from the Environment*. San Francisco: W. H. Freeman.

[von Ahn 2004] von Ahn, L., and Dabbish, L., "Labeling Images with a Computer Game." ACM CHI.

[Wang 1995] Wang, X., "Learning by Observation and Practice: An Incremental Approach for Planning Operator Acquisition." *Proceedings of the Twelfth International Conference on Machine Learning*. San Francisco: Morgan Kaufmann.

[Weizenbaum 1976] Weizenbaum, J. *Computer Power and Human Reason*. W.H. Freeman and Company.

[Zettlemoyer 2005] Zettlemoyer, L., Pasula, H., and Kaelbling, L., "Learning Planning Rules in Noisy Stochastic Worlds." *Proceedings of the Twentieth National Conference on Artificial Intelligence* (AAAI 2005).