Semantic Classification of Utterances in a Language-driven Game

Kellen Gillespie[1,2], Michael W. Floyd[2], Matthew Molineaux[2],
Swaroop S. Vattam[3], and David W. Aha[4]

[1]Amazon.com, Inc.; Seattle, WA; USA
kelleng@amazon.com
[2]Knexus Research Corporation; Springfield, VA; USA
michael.floyd@knexusresearch.com
matthew.molineaux@knexusresearch.com
[3]MIT Lincoln Laboratory (Group 52); Lexington, MA; USA
swaroop.vattam@ll.mit.edu
[4]Naval Research Laboratory (Code 5514); Washington, DC; USA
david.aha@nrl.navy.mil

Abstract. Artificial agents that interact with humans may find that understanding those humans' plans and goals can improve their interactions. Ideally, humans would explicitly provide information about their plans, goals, and motivations to the agent. However, if the human is unable or unwilling to provide this information then the agent will need to infer it from observed behavior. We describe a goal reasoning agent architecture that allows an agent to classify natural language utterances, hypothesize about a human's actions, and recognize their plans and goals. In this paper we focus on one module of our architecture, the *Natural Language Classifier*, and demonstrate its use in a multiplayer tabletop social deception game, *One Night Ultimate Werewolf*. Our evaluation indicates that our system can obtain reasonable performance even when the utterances are unstructured, deceptive, or ambiguous.

Keywords: Semantic classification, social deception game, tabletop game, goal reasoning

1 Introduction

Agents that interact with humans, cooperatively or competitively, can benefit from understanding those humans' plans and goals. By having this information, the agent can more effectively assist a human teammate or thwart an adversarial human. While in some circumstances a human may directly and concisely provide its plans and goals, it is often more realistic that the agent will need to infer this information based on the human's behavior. In this work, we consider a particular problem domain where humans do not unambiguously share this type of information, and will often attempt to intentionally conceal it through deception.

In this paper, we describe our architecture for an agent that classifies natural language utterances to hypothesize about humans' plans and goals. We have previously shown that such an agent can successfully predict squad members' goals in a military domain (Gillespie et al., 2015). However, deploying the agent in a social deception game adds the following complexities:

- **Human cooperation**:
  - *Military domain*: The humans are squad members working in collaboration with the agent.
  - *Social deception game*: The humans can be teammates of the agent but can also be neutral or adversaries.
- **Language**:
  - *Military domain*: The fixed-vocabulary language is highly constrained.
  - *Social deception game*: There are minimal constraints on the language.
- **Clarity of utterances**:
  - *Military domain*: The utterances will be direct, concise, and unambiguous.
  - *Social deception game*: The utterances may be incomplete, ambiguous, incorrect, or deceptive. Additionally, some utterances may have no relevance to the game (e.g., casual conversation among players).

Although our focus has been on military scenarios and social deception games, the ability to reason about goals from natural language is also relevant in other domains such as those involving negotiations, diplomacy, and legal reasoning.

While we describe the entire agent architecture in Section 2, our focus in this paper is on the module that allows the agent to classify the semantic meaning of each utterance. Section 3 provides an introduction to the social deception game we use, One Night Ultimate Werewolf, and Section 4 presents our approach for extracting information from in-game utterances. In Section 5, we describe an evaluation using logs of actual gameplay and show that the agent is able to classify several key aspects of each utterance. We examine related work in Section 6 and present future research directions in Section 7.

2 Agent Architecture

Our agent interprets and responds to its environment via a five-step goal reasoning process (Klenk et al., 2013; Aha, 2015). This process allows an agent to dynamically refine its goals in response to unexpected external events or opportunities, and enact plans to accomplish those goals. The agent's decision cycle is shown in Fig. 1 and has five primary components:

1. **Natural Language Classifier**: This module listens for natural language *utterances* (i.e., spoken language) in the environment and attempts to extract semantic meaning from the utterances. For each utterance received, the module outputs a multi-label *classification of the utterance*.

2. **Explanation Generator**: This module uses the *classified utterances* and *environmental observations* (i.e., the current state of the environment) to generate possible explanations for what has occurred in the environment (Molineaux and Aha, 2015). The explanation contains, in part, the agent's hypothesis as to what actions each other entity (e.g., humans, robots, or other agents) in the environment must have performed for the environment to have changed from its prior state to the current state. As more classified utterances and state observations are received, the Explanation Generator further refines its explanation. The most likely *actions* for each entity are output.
3. **Plan Recognizer**: For each entity in the environment, the Plan Recognizer receives a sequence of *actions* that the entity may have performed (i.e., one action in the sequence every time the Explanation Generator produces output). The Plan Recognizer uses the sequence of actions to identify the entity's plan (Vattam et al., 2014). The Plan Recognizer assumes that each plan achieves a goal, so the recognized plan can be used to identify the entity's current goal. This module outputs the recognized *goal* of each entity in the environment.
4. **Goal Selector**: This module monitors for any changes in the *goals of the entities* or external events, and can modify the agent's goal in response. This allows the agent to dynamically respond to any unexpected behaviors or opportunities (i.e., the agent changes its goal to better respond to other entities' goals). The output of this module is the *agent's goal* (even if the goal is unchanged).
5. **Plan Generator**: If the *agent's goal* has changed, the Plan Generator generates a new plan for the agent to perform. The plan generator also monitors the progress of the current plan to determine if it is necessary to repair the plan or generate a new plan. The output of this module are the *actions* (of the plan) that the agent is attempting to perform.

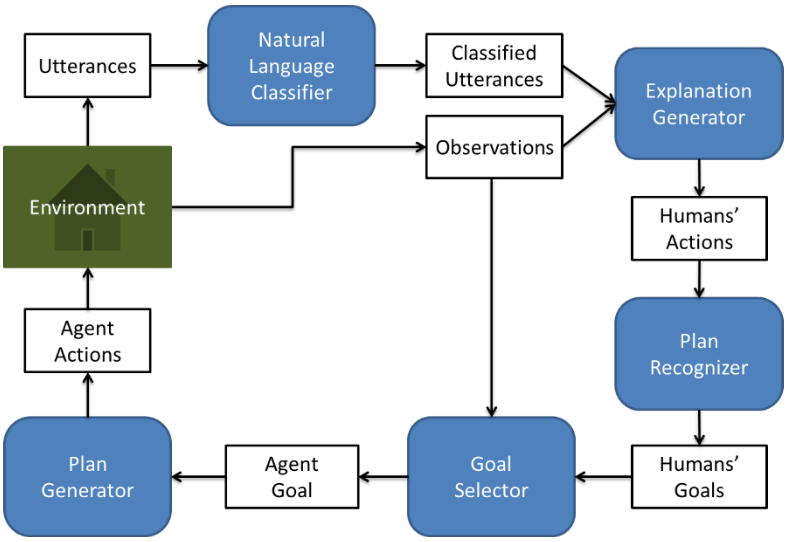

Figure 1: Decision cycle of the agent

In this paper we focus exclusively on the Natural Language Classifier and how it generates classified utterances from unconstrained natural language.

3 Background: One Night Ultimate Werewolf

The domain we are examining is a tabletop social deception game called *One Night Ultimate Werewolf*[1] (Bezier Games, 2016). We chose Ultimate Werewolf because players interact using unconstrained natural language, have a variety of goals, work under hidden information, and actively engage in deception.

In the game, players are randomly assigned *roles* that place them into three competing factions: *Villagers*, *Werewolves,* and the *Tanner*. The goal of the Villagers is to identify which players are Werewolves, the goal of the Werewolves is to avoid detection, and the goal of the Tanner is to convince the Villagers that it is a Werewolf. We constrained the game to five players and eight possible roles (i.e., five roles will be assigned and three will be unused), with some roles granting special abilities. The roles we use are: *Werewolf* (x2), *Mason* (x2), *Generic Villager* (x2), *Seer*, and *Tanner*. The Werewolf roles are part of the *Werewolves faction*, the Tanner is part of the *Tanner faction*, and all remaining roles are part of the *Villagers faction*. The three unused role cards are placed, face down, on the table.

The game proceeds as follows:

1. **Role assignment:** Each player receives a *role card* with an assigned role printed on it. After viewing their role, the player then places the card face down in front of them. They may not view their card again[2].
2. **Special abilities:** An external moderator oversees this portion of the game:
   (a) The moderator instructs all players to close their eyes.
   (b) The moderator instructs all Werewolves to open their eyes, identify the other Werewolves (if any), and close their eyes. If only one Werewolf opens their eyes, they may look at one of the unused role cards.
   (c) The moderator instructs all Masons to open their eyes, identify the other Masons (if any), and close their eyes.
   (d) The moderator instructs the Seer to open their eyes. The Seer may look at the role card of one other player or two of the unused role cards. The Seer then closes their eyes.
   (e) The moderator instructs all players to open their eyes again.
3. **Information gathering**: The players have several minutes to attempt to gather information about the other players. There is no turn-taking so players can speak as much or as little as they wish. Similarly, there are no constraints on what is discussed or the vocabulary used.

[1] We will refer to the game as *Ultimate Werewolf* for the remainder of the paper.

[2] Although viewing your role again does not influence our game, in some versions of Ultimate Werewolf a player's role can be switched without their knowledge.

4. **Shooting phase**: Each player chooses one other player to "shoot" and players announce their choices simultaneously. The player who is shot by the most other players "dies". In the event of a tie, all players tied for the most shots die.
5. **Declaring winners**:
   (a) If the Tanner dies, the Tanner wins (regardless of which other players die). Otherwise, the Tanner loses.
   (b) If at least one Werewolf dies, the Villagers faction wins (regardless of the Tanner's fate). Otherwise, they lose.
   (c) If the Tanner does not die and no Werewolves die, the Werewolves faction wins. Otherwise, the Werewolves lose.

Each player knows their own role and, depending on their special ability, may have more information as well (i.e., from special abilities). The Werewolves and Masons know information about other members of their faction; the Seer may know the role of any one other player; and a lone Werewolf or the Seer may know either 1 or 2 unused roles. Players with the Generic Villager role have no special abilities, so they have less information than other players.

4 Multilabel and Multiclass Semantic Classification

The Natural Language Classifier receives as input each natural language *utterance* that it can sense in the environment. Each utterance represents a continuous unit of speech with a distinct beginning and ending (e.g., *"I think you are a werewolf."* or *"Did you look at anyone's role?"*). Utterances are encoded using a bag-of-words representation. An utterance u is a set containing each word w in the utterance:

$$u = \{w_a, w_b, \dots\}$$

For example, "*I think you are a werewolf.*" would be represented as $\{'I', 'think', 'you', 'are', 'a', 'werewolf'\}$. We classify each utterance along nine different dimensions using a set of parallel classifiers. The classification tasks and their associated class labels are listed below:

- **Purpose**: The general type of utterance being made.
  - **Classes**: *claim* (make a factual claim), *question* (ask a question), *hypothesis* (pose a hypothesis), *suggest-target* (suggest a target to shoot), *self-explain* (explain the player's behavior to the group), *other* (an utterance that does not fall under any of the other classes).
- **Address-type**: The size of the group the utterance was addressed to.
  - **Classes**: *everyone* (the utterance was directed at all or most of the players), *one-person*, *two-people*
- **Addressee**: Whether an utterance is directed to a specific player. This classification task is complementary to Address-type (i.e., a *known* Addressee only occurs when the Address-type is *one-person* or *two-people*).
  - **Classes**: *known* (the utterance directly addresses one of the players), *none* (no specific player is addressed)

- **Subject**: The subject matter discussed in the utterance.
  - **Classes**: *starting-role* (a player's role when they viewed their role card), *unused-role* (roles that were not assigned to anyone), *starting-role-group* (a subgroup of possible roles for a player), *role-observe-performer* (whether a player has a role that allows the observation of other players' roles), *role-observe-target* (whether a player had their role observed by another player), *divulge* (a player provides information about themselves to other players), *statement* (the utterance is in regards to a previously made statement), *shoot-target* (discusses targeting a player for shooting)
- **Target-role**: The role being discussed in the utterance.
  - **Classes**: *none* (no role is being discussed), *unknown* (a role is being discussed but the exact role is not known), *Seer*, *Werewolf*, *Villager*, *Mason*, *Tanner*.
- **Target-role-group**: The subgroup of roles is being directly discussed.
  - **Classes**: *none*, *villagers*, *non-villagers*, *paired-roles* (roles, either Masons or Werewolves, that can view the other members with the same role).
- **Target-player**: The player being discussed in the utterance.
  - **Classes**: *known* (directly referring to one of the players), *unknown* (a player is discussed but the exact player is unknown), *none* (no player is discussed).
- **Target-position**: The presence and location of an unused role card on the table (e.g., a card viewed by the Seer, knowledge of an unused role because there were no other Werewolves).
  - **Classes**: *one-unknown* (a role is unused but its position is unknown), *two-unknown* (two roles are unused but their positions are unknown), *three-unknown* (three roles are unused but their positions are unknown), *left* (the leftmost unused role card), *middle* (the middle unused role card), *right* (the right unused role card), *none* (no unused role is mentioned).
- **Negation**: Whether a statement is positive (e.g., something happened or is true) or negative (e.g., something did not happen or is not true).
  - **Classes**: *positive*, *negative*

4.1 Classifiers

We examine three methods for training the classifiers used by the Natural Language Classifier: *Frequency*, *Probabilistic*, and *Probabilistic Frequency*. All three methods use a dictionary of known words. If there are N known words, the dictionary $dict$ will contain N entries ($dict = \langle w_1, w_2, \dots, w_N \rangle$). Each utterance u is filtered to remove stop words and converted to a vector v_u of length N ($v_u = \langle m_1, m_2, \dots, m_N \rangle$). The *i*th element in v_u (i.e., m_i) contains the multiplicity in the utterance of the i^{th} element in $dict$ (i.e., w_i). For example, if the 3rd word in the dictionary is '*werewolf*' and the word '*werewolf*' occurred in the utterance once, the 3rd element of v_u would be 1.

The three classification methods learn *classification vectors* from a set of labelled training utterances. Like the utterance vectors, the classification vectors are of length N (i.e., classification vector $cv = \langle s_1, s_2, \dots, s_N \rangle$). For each classification task, the training examples are partitioned by class and one classification vector is learned for each class (e.g., for the *Negation* task the training examples are partitioned into one set with the *positive* label and one set with the *negative* label). The three methods generate classification vectors as follows:

Frequency

All utterance vectors from a partition are summed. If the utterance vectors from class C are in partition p_C, classification vector cv_C^{freq} for that class is:

$$cv_C^{\text{freq}} = \sum_{v_{u_i} \in p_C} v_{u_i}$$

Since each utterance vector encodes the number of times each word appeared in the utterance, the classification vector contains the total number of times each word appeared for a given class.

Probabilistic

The Probabilistic classification vector cv_C^{prob} is computed by dividing each element of the Frequency classification vector by the number of utterances in the partition:

$$cv_C^{prob} = \frac{cv_C^{freq}}{|p_C|}$$

This classification vector represents what percentage of utterances in the partition contained each word.

Probabilistic Frequency

The Probabilistic Frequency classification vector cv_C^{pf} is calculated using both the Frequency and Probabilistic classification vectors. A new classification vector is created such that the ith element is the product of the ith elements in the Frequency and Probabilistic classification vectors:

$$cv_C^{pf} = \langle s_{C,1}^{freq} \times s_{C,1}^{prob}, s_{C,2}^{freq} \times s_{C,2}^{prob}, \ldots, s_{C,N}^{freq} \times s_{C,N}^{prob} \rangle$$

4.2 Classification

An input utterance is classified by the Natural Language Classifier using the learned classification vectors. If a classification task *l* has a set of possible labels C_l, the Natural Language Classifier computes the dot product between the utterance vector and each of the classification vectors for that classification task (e.g., to find the *Negation* classification, only the classification vectors for the *positive* and *negative* classes are used). The associated label of the classification vector that maximizes that value is assigned to the utterance:

$$label_l = \underset{C_i \in \mathcal{C}_l}{\operatorname{argmax}} \, v_u \cdot cv_{C_i}$$

In the Ultimate Werewolf domain, nine labels are assigned to each input utterance.

5 Evaluation

In our empirical evaluation we assess whether the agent can correctly classify natural language utterances using multilabel and multiclass semantic classification. Using data from real games of Ultimate Werewolf, our results show that our agent can extract important semantic information from utterances without limiting the language of players.

5.1 Data Collection

We collected data from eight games of Ultimate Werewolf, with each game being played by five human players. The same five players participated in all eight games. In addition to the rules described in Section 3, the players were also encouraged to use proper names when referring to each other. This was done because the agent only has access to the audio of the game (i.e., it cannot see who a player is facing when speaking). However, this was not strictly enforced so there are instances where the players use pronouns. No other limitations were placed on vocabulary, utterance structure, conversation ordering, or topics of discussion.

Audio was recorded for each game along with the players' roles, special ability actions (e.g., if they viewed another player's role), and shooting targets. Each recording was manually transcribed and separated into the individual utterances. The mean number of utterances per game was 49.1, with a minimum of 36 and a maximum of 69. Each utterance was manually labelled for each of the nine classification tasks. The labelling was done by a third party (i.e., not the players themselves), so it represents how an external observer would classify each utterance rather than a player's intended meaning (e.g., how the observer interpreted ambiguous statements).

5.2 Experimental Setup

Evaluation was performed using leave-one-out testing (i.e., each run used seven annotated game transcripts for training and one for testing). The utterances from the testing transcript were given as input to the agent. The performance of the agent (i.e., how well its classification matched the annotated classes of the utterance) was measured for each of the nine classification tasks. We used the *F_1 score* to measure performance ($F_1 = 2\,\frac{precision \times recall}{precision + recall}$). The three classification methods described in Section 4 were evaluated: *Frequency*, *Probabilistic*, and *Probabilistic Frequency*. The results from these three classification approaches were also compared to a baseline that randomly classifies each utterance (referred to as *Random* in our results).

5.3 Results

The results for each of the nine classification tasks and the overall performance are shown in Fig. 2 and Fig. 3. The Probabilistic and Probabilistic Frequency approaches outperformed the baseline over all classification tasks and outperformed the Frequency approach over all tasks except Target-role-group (i.e., all three approaches achieved similar results for this task). Other than the Target-role task (where Probabilistic Frequency performed better), and Purpose and Target-role-group (where they performed similarly), the Probabilistic method outperformed the Probabilistic Frequency method. The Frequency approach performed poorly, underperforming the Random baseline in six of the classification tasks and recording a lower average F_1-score.

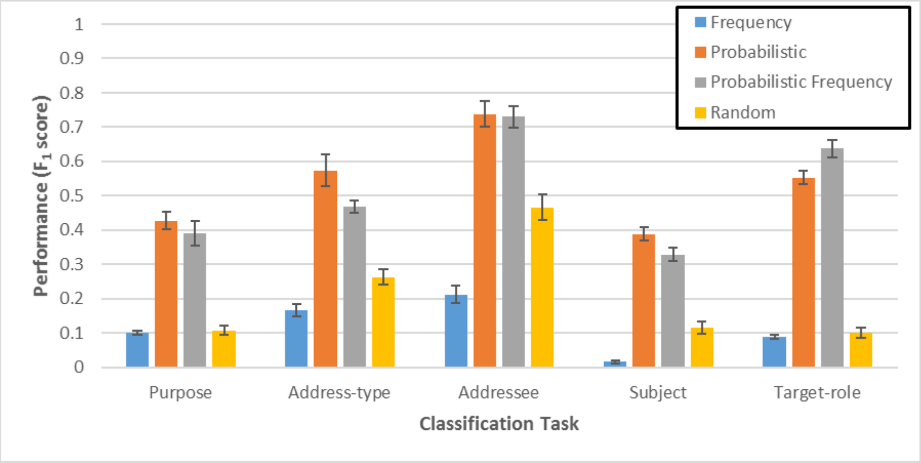

Figure 2: Classification performance for the *Purpose*, *Address-type*, *Addressee*, *Subject*, and *Target-role* tasks

5.4 Discussion

The classification tasks have between two and eight classes each (with a median of 4). We observed an inverse correlation between the number of classes and agent performance. The two classification tasks that do not follow this inverse correlation are the Target-role and Target-role-group tasks. Target-role has seven classes but the agent performed better than expected on this task. The primary reason for this is because the utterances contain keywords (i.e., the name of the role) that make them easy to classify. In contrast, the agent performed poorly on the Target-role-group task, which has only four classes. This is because the agent has difficulty determining if an utterance is explicitly discussing one of the groups or only implicitly referencing the group by mentioning one of the roles in that group. This is especially prevalent since the players use group names that are similar to role name. For example, "*I think you are one of the villagers*" would be classified as *villagers* (i.e., it discusses the villagers group) whereas "*I think you are the Villager*" would be classified as *none* (i.e., a role is discussed, not an entire group).

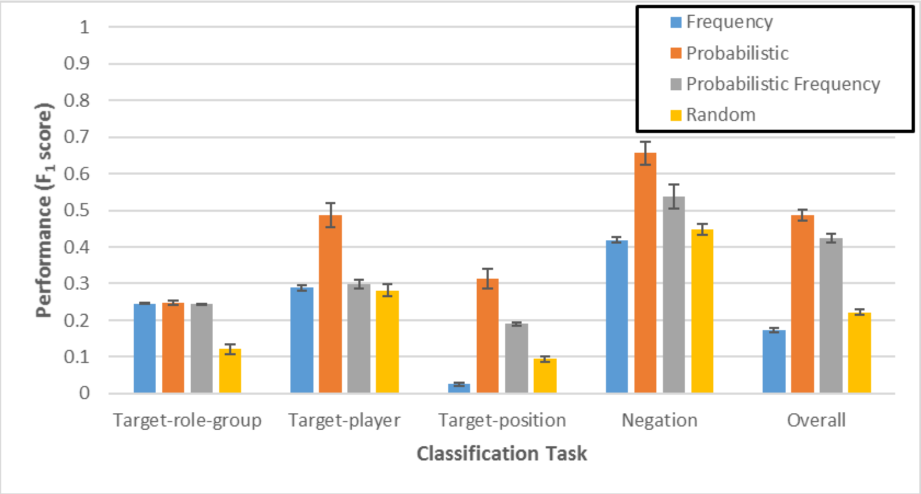

Figure 3: Classification performance for the *Target-role-group*, *Target-player*, *Target-position*, *Negation*, and *Overall* tasks

The classes are highly imbalanced given the wide range of possible utterances. In our dataset, between 45% and 96% of utterances belong to the majority class ($\mu = 69\%$) and between 0.5% and 28% of the utterances belong to the least frequent class ($\mu = 7\%$). While this imbalance affects all three classification methods, it is the primary reason the Frequency method performs poorly. For each class, the Frequency method counts the number of times each word appears in the training examples. This causes classes with more training examples (i.e., the majority class) to have higher frequency values and therefore be more likely to be the labelled class of an input utterance. Even if a specific word is a strong indication that an utterance should be labelled as the minority class, if that word appears occasionally in the majority class it can cause the classifier to label the utterance as the majority class. The Probabilistic and Probabilistic Frequency approaches help mitigate the class imbalance problem by taking into account the percentage of training examples that contain each word rather than just the number of times a word occurs. However, as with the Frequency approach, they also suffer from having very few training examples for some classes (e.g., some classes only have a single example in the dataset). Additionally, some classes have such a wide range of different utterances (e.g., non-game talk amongst the players) that it makes it difficult to learn a model for that class even if a significant number of examples are available.

Our results, while an improvement over the baseline, fall well short of ideal performance. Given the difficulty of the problem (i.e., unconstrained text, rapid changes in topics, highly unbalanced data, ambiguity), we expected the agent to have difficulty classifying the utterances but are unsure what performance is necessary for the remaining components (i.e., how erroneous the classifications can be before the Explanation Generator and Plan Recognizer fail). Even for a human annotator, the utterances were often highly ambiguous and difficult to classify. While the agent should ideally accurately predict all nine categories, it may be possible that the remaining modules can achieve reasonable results even if only a subset of each utterance's classifications are correct. We intend to investigate the system's sensitivity to classification performance in future work.

As was shown in our results, the Probabilistic method achieved the best performance on most tasks but Probabilistic Frequency performed best on the Target-role classification task. This indicated that it will likely be necessary to determine the best performing classification strategy on each task or use an ensemble approach rather than committing to a single strategy for all tasks. Given our current level of performance, this will also necessitate exploring new classification approaches and taking steps to manage the class imbalance problem (e.g., collect more data, balance the dataset, use label regularization (Mann and McCallum, 2007)).

6 Related Work

Our work focuses on utterance classification in a game where the players often engage in deception. Although we do not attempt to identify which utterances or players are deceptive, related work in deception detection often addresses similar problems. Deception detection in conversational games has been approached using textual cues (Zhou and Sung, 2008) (e.g., word selection, utterance duration, utterance complexity), vocal cues (Chittaranjan and Hung, 2010) (e.g., pitch, pauses, laughter), and visual cues (Raiman et al., 2011) (e.g., head and arm movements). These systems are designed to classify players as truthful or deceptive, and use that information to identify players with deceptive roles (e.g., werewolves). However, while collecting experimental data we observed that even players with roles that should not require deception (e.g., villagers) actively engage in deception and omission. Since nearly all players engage in deception, it becomes more important to identify when they are being deceptive and why they are being deceptive.

Network analysis has been used to identify groups of players with similar patterns of behavior (Yu et al., 2015). The statements made by each player are used to determine their attitudes toward other players (e.g., a positive attitude if they regularly defend another player or a negative attitude if they regularly accuse another player) and players are clustered based on their attitudes. The underlying assumption is that deceptive players will have positive attitudes toward other deceptive players while having negative attitudes toward other players. In our domain, even the most common roles (e.g., Werewolf, Mason, Generic Villager) only have at most two players with those roles. If a player knows of another player with the same role (i.e., using a special ability), they often avoid displaying a positive attitude toward that player since it can arouse suspicion.

Azaria et al. (2015) have developed an agent that is able to identify deception, convince other players of the deception, and avoid raising suspicions about their own behavior. The agent participates in a simplified social deception game where a single pirate has to deceive three non-pirates in order to steal treasure. The primary differences between their work and our own are that their game uses structured sentences rather than free text, the game is less complex (i.e., fewer roles and player goals), and their system is focused on identifying deception rather than a player's plan or role.

Orkin and Roy (2010) use sequences of utterances and actions to predict a player's behavior in a restaurant simulation game. Due to the number of utterances possible using free-form text, they had relatively poor performance when training with 8-10 game logs compared to 30-100 game logs. This is similar to our own evaluation where many of the classes had few training instances. They found that increasing the number of training logs increased performance but required significant annotation time (approximately 56 hours). In the AutoTutor Intelligent Tutoring System (Olney et al., 2003), utterances are used to determine when initiative has changed and determine the needs of the student. For example, certain utterances indicate the student has switched from providing responses to being stuck or asking questions. This can be thought of as a simplified version of plan recognition, where the student has three plans: *respond*, *ask questions*, or *do nothing*. However, only a single utterance is used for each classification, rather than the entire sequence of utterances.

Vázquez et al. (2015) have studied the reaction of human players when a robotic player participates in a social deception game. The robot has the appearance of autonomy but is actually controlled by an unseen human. Although this differs from our own goal of an autonomous player, it does demonstrate that humans are open to playing social deception games with robotic participants.

7 Conclusions and Future Work

We described our architecture for an agent that uses unstructured natural language utterances to reason about the plans and goals of humans. In this paper, we focus on one module of this architecture, the Natural Language Classifier, and examine its ability to classify utterances in a multiplayer tabletop social deception game. Our previous work (Gillespie et al., 2015) described the application of our agent architecture in a military domain. However, in this paper we chose to examine a social deception game because it posed several interesting challenges, including less constrained language, deception, and ambiguity.

The Natural Language Classifier extracts information from each utterance by assigning labels according to nine distinct classification tasks. We studied its ability using three supervised learning methods for these tasks. We evaluated it in the social deception game Ultimate Werewolf using logs of eight games played by human players. We found that classification that considers only word frequency performed poorly, whereas the other two classification methods achieved reasonable results and outperformed our baseline.

Our principal area of future work is to integrate the Natural Language Classifier with the other components of the agent architecture and evaluate the agent's overall performance. We performed such an evaluation in a military domain, but performing this integration for Ultimate Werewolf will require a better understanding of the minimum performance necessary during utterance classification. Currently, we have a limited corpus of training data that was collected from a single set of players. Different players are likely to use different utterances and a different vocabulary, so it will be important to collect data from a variety of players. Additionally, we plan to allow the agent to observe games of Ultimate Werewolf and make predictions about player roles, identify deception, and learn the motivations of individual players.

References

Aha, D.W. (Ed.) (2015). *Goal Reasoning: Papers from the ACS Workshop* (Technical Report GT-IRIM-CR-2015-001). Atlanta, USA: Georgia Institute of Technology, Institute for Robotics and Intelligent Machines.

Azaria, A., Richardson, A., & Kraus, S. (2015). An agent for deception detection in discussion based environments. In *Proceedings of the Eighteenth ACM Conference on Computer Supported Cooperative Work & Social Computing* (pp. 218-227). Vancouver, Canada: ACM.

Bezier Games. (2016). *One night ultimate werewolf.* Retrieved from [beziergames.com/collections/all-games/products/one-night-ultimate-werewolf]

Chittaranjan, G., & Hung, H. (2010). Are you a werewolf? Detecting deceptive roles and outcomes in a conversational role-playing game. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing* (pp. 5334-5337). Dallas, USA: IEEE.

Gillespie, K., Molineaux, M., Floyd, M.W., Vattam, S.S., & Aha, D.W. (2015). Goal reasoning for an autonomous squad member. In D.W. Aha (Ed.) *Goal Reasoning: Papers from the ACS Workshop* (Technical Report). Atlanta, USA: Georgia Institute of Technology, Institute for Robotics and Intelligent Machines.

Klenk, M., Molineaux, M., & Aha, D.W. (2013). Goal-driven autonomy for responding to unexpected events in strategy simulations. *Computational Intelligence*, *29*(2), 187-206.

Mann, G.S, & McCallum, A. (2007). Simple, robust, scalable semi-supervised learning via expectation regularization. In *Proceedings of the Twenty-Fourth International Conference on Machine Learning* (pp. 593-600), Corvallis, USA: ACM.

Molineaux, M., & Aha, D.W. (2015). Continuous explanation generation in a multi-agent domain. In *Proceedings of the Third Conference on Advances in Cognitive Systems*. Atlanta, USA: Cognitive Systems Foundation.

Olney, A.M., Louwerse, M., Matthews, E., Marineau, J., Hite-Mitchell, H., & Graesser, A.C. (2003). Utterance classification in AutoTutor. In *Proceedings of the Workshop on Building Educational Applications Using Natural Language Processing at the Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics*. Edmonton, Canada.

Orkin, J., & Roy, D. (2010). Semi-automated dialogue act classification for situated social agents in games. In *Proceedings of the Agents for Games & Simulations Workshop at the Ninth International Conference on Autonomous Agents and Multiagent Systems*. Toronto, Canada.

Raiman, N., Hung, H., & Englebienne, G. (2011). Move, and I will tell you who you are: Detecting deceptive roles in low-quality data. In *Proceedings of the Thirteenth International Conference on Multimodal Interfaces* (pp. 201-204). Alicante, Spain: ACM.

Vattam, S.S., Aha, D.W., & Floyd, M. (2014). Case-based plan recognition using action sequence graphs. In *Proceedings of the Twenty-Second International Conference on Case-Based Reasoning* (pp. 495-510). Cork, Ireland: Springer.

Vázquez, M., Carter, E.J., Vaz, J.A., Forlizzi, J., Steinfeld, A., & Hudson, S.E. (2015). Social group interactions in a role-playing game. In *Proceedings of the Tenth Annual ACM/IEEE International Conference on Human-Robot Interaction* (pp. 9-10). Portland, USA: ACM.

Yu, D., Tyshchuk, Y., Ji, H., & Wallace, W.A. (2015). Detecting Deceptive Groups Using Conversations and Network Analysis. In *Proceedings of the Fifty-Third Annual Meeting of the Association for Computational Linguistics* (pp. 857-866). Beijing, China: ACL.

Zhou, L., & Sung, Y.-W. (2008). Cues to Deception in Online Chinese Groups. In *Proceedings of the Forty-First Hawaii International International Conference on Systems Science*. Waikoloa, USA.