Detecting Deceptive Groups Using Conversations and Network Analysis

Dian Yu[1], Yulia Tyshchuk[2], Heng Ji[1], William Wallace[2]

[1]Computer Science Department, Rensselaer Polytechnic Institute
[2]Department of Industrial and Systems Engineering, Rensselaer Polytechnic Institute
[1,2]{yud2,tyshcy,jih,wallaw}@rpi.edu

Abstract

Deception detection has been formulated as a supervised binary classification problem on single documents. However, in daily life, millions of fraud cases involve detailed conversations between deceivers and victims. Deceivers may dynamically adjust their deceptive statements according to the reactions of victims. In addition, people may form groups and collaborate to deceive others. In this paper, we seek to identify deceptive groups from their conversations. We propose a novel subgroup detection method that combines linguistic signals and signed network analysis for dynamic clustering. A social-elimination game called *Killer Game* is introduced as a case study[1]. Experimental results demonstrate that our approach significantly outperforms human voting and state-of-the-art subgroup detection methods at dynamically differentiating the deceptive groups from truth-tellers.

1 Introduction

Deception generally entails messages and information intentionally transmitted to create a false conclusion (Buller et al., 1994). Deception detection is an important task for a wide range of applications including law enforcement, intelligence gathering, and financial fraud. Most of the previous work (*e.g.*, (Ott et al., 2011; Feng et al., 2012)) focused on content analysis of a single document in isolation (*e.g.*, a product review). The promoters of a product may post fake complimentary reviews, while their competitors may hire people to write fake negative reviews (Ott et al., 2011).

However, when we want to detect deception from text or voice conversations, the deception behavior may be affected by the following factors beyond textual statements.

1. *Dynamic*. Recent research in social science suggests that deception communication is dynamic and involves interactions among people (*e.g.*, (Buller and Burgoon, 1996)). Additionally, the research postulates that human's capacity to learn by observation enables him to acquire large, integrated units of behavior by example (Bandura, 1971). Therefore, a person's behavior concerning deception or truth-telling can change constantly, while he learns from others' statements during conversations.
2. *Global*. People may form groups for purpose of deception. Research in social psychology has shown that an individual's object-related behavior may be affected by the attitudes of other people due to group dynamics (Friedkin, 2010).

Recent studies typically have been conducted over "*static*" written or oral deceptive statements. There is no obligatory requirement for communication between the author and the readers of these statements (Yancheva and Rudzicz, 2013). As a result, a victim of deception tends to trust the story mainly based on the statement he reads (Ott et al., 2011). However, in daily life, millions of fraud cases involve detailed conversations between deceivers and victims. A deceiver may make a statement, which is partially true in order to deceive or mislead victims and adjust his deceptive strategies based on the reactions of victims (Zhou et al., 2004). Therefore, it is more challenging to identity a deceiver in an interactive process of deception.

Most deception detection research addressed individual deceivers, but deceivers often act in pairs or larger groups (Vrij et al., 2010). The interac-

[1]The data set is publicly available for research purposes at: http://nlp.cs.rpi.edu/data/killer.zip

Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing, pages 857–866, Beijing, China, July 26-31, 2015.

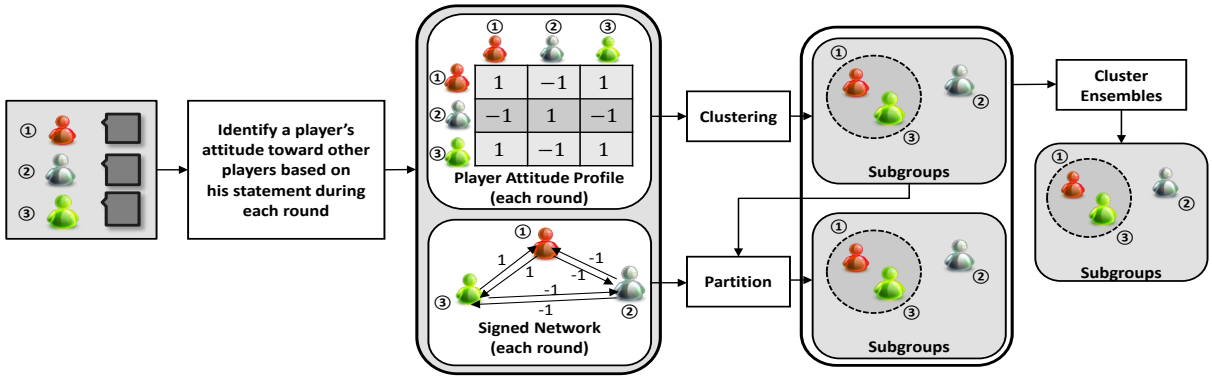

Figure 1: Deceptive group detection for a single round.

tions within a deceptive group have been ignored. For example, a product review from a deceiver may be supported by his teammates so that his deceptive comments can be read by more potential buyers. In this case, we can identify a deceptive group based on their collaborations and common characteristics, which is more promising than the typical methods of classifying individual statements as deceptive or trustworthy.

In order to identify deceptive groups by analyzing the evolution of a person's deception strategy during his interactions with victims and the interactions within the deceptive group from conversations, we use a social-elimination game called *Killer Game* which contains the ground-truth of subgroups.

The killer game has many variants that involve different roles and skills. We choose a classical version played by three roles/teams: detectives, citizens, and killers. The role of each *player* (game participant) is randomly assigned by a third-party game judge. Every killer/detective is given the identities of his teammates. There are two alternating phases of the game: "*night*", when killers may covertly "*murder*" a player and detectives may learn one player's role; and "*day*", when surviving players are informed of who was killed last "*night*" and then asked to speculate about the roles of other surviving players. Before a "*day*" ends, every surviving player should vote for a suspect. The candidate with the most votes is eliminated. A player's identity is not exposed after his "*death*". The game continues until all killers have been eliminated or all detectives have been killed. The killers are treated as deceivers, and citizens and detectives as truth-tellers.

In this paper, we present an unsupervised approach for differentiating the deceptive groups from truth-tellers in a game. During each round, we use Natural Language Processing (NLP) techniques to identify a player's attitude toward other players (Section 2), which are used to construct a vector of attitudes for each surviving player (Section 3.1) and a signed social network representation (Section 3.2) for the discussions. Then we use a clustering algorithm to cluster the attitude vector space and obtain results for each round (Section 3.1). We also implement a greedy optimization algorithm to partition the singed network based on the attitude clustering result (Section 3.2). Finally, we apply a pairwise-similarity approach that makes use of the predicted co-occurrence relations between players to combine all results from each round (Section 3.3). Figure 1 provides an overview of our system pipeline.

The major novel contributions of this paper are as follows.

- This is the first study to investigate conversations and deceptive groups for computerized deception detection.
- The proposed clustering technique is shown to be successful in separating deceptive groups from truth-tellers.
- The method can be applied to dynamically detect subgroups in a network with discussants who tend to change their opinions.

2 Attitude Identification

In this section, we describe how we take a player's statement in a single round as input to extract his attitudes toward other players and represent them by an attitude 3-tuple (speaker, target, polarity) list. For this work, the polarity of attitudes (Balahur et al., 2009) can be positive (1), negative (-1) or neutral (0). A game log from a single round

will be used as our illustrative example, as shown in Figure 2.

C: CITIZEN; D: DETECTIVE; K: KILLER

System: First Round.
System: 15 was killed last night. 15, please leave your last words.
15(C): I'm a citizen. Over.
16(K): I'm a good person. 11 and 2 are suspicious.
1(K): I'm a good person. It has been a long time since I played as a killer. I'm a citizen. 11 is suspicious and I don't want to comment on 16's statement.
2(C): I'm a detective. 6 was proved as a killer last night. Over.
3(C): I don't know 2's identity. It's hard to judge 16's statement. 1 seems to be a good person. I'm a citizen.
4(C): Citizen. I cannot find a killer. I trust 2 since 2 sounds a good person. 16 is suspicious. I regard 16 as a killer. I'm 2's teammate.
5(D): I'm a detective. I verify 2's identity and 2 is a killer. 13 is good.
6(C): Why do you want to attack 2? I don't understand. 14 is suspicious.
7(K): It's hard to define 6's identity. 4 may be a citizen. I will vote for 2. 6 sounds very weird and I found 6 very suspicious. I will follow the detective 5 to vote for 2.
8(C): We should calm down. 7 seems to be a bad person.
9(C): 1 and 7 seem to be killers. There is no evidence to support 2 as a detective. 3 is a citizen. 4 is possibly a detective. 6 is also good.
10(D): I agree with you. 7 must be a killer. 2 and 7 should debate.
11(C): I don't know 2 but I think 2 is good. 3 is good. There should be one or two killers among 1, 4 and 7.
12(K): 11 sounds like a killer. 2 is a killer. I'm a citizen. Vote for 2.
13(D): 15 is a citizen. 16 is logically good. I think 1, 8, 9, 10 are OK. I don't think 2 is a killer. I doubt 7's intention. Please vote for 7.
14(D): 10, 13, 16 are good. I don't think 7 must be a killer. 2 is obviously bad. I'm a citizen.
System: 16, 11, 14, 7, 1, 3, 8, 12, 4 vote for 2 $\cdots$ 10, 13, 5, 2 vote for 7 $\cdots$ 9, 6 vote for 11 $\cdots$ *2 is out.*

Figure 2: Killer game sample log (the 1st round).

2.1 Target and Attitude Word Identification

We start by identifying targets and attitude words from conversations. In the killer game, a target is represented by his unique ID[2] and game terms are regarded as attitude words. We collected 41 terms in total from the game's website [3] and related discussion forum posts. ICTCLAS (Zhang et al., 2003) is used for word segmentation and part-of-speech (POS) tagging. There are two kinds of game terms: positive and negative. Positive terms include "*citizen*", "*good person*", "*good person certified by the detectives*" and "*detective*". Negative terms include "*killer*", "*killer verified by the detectives*" and "*a killer who claimed himself/herself to be a detective*". We assign the polarity score +1, -1 to positive and negative terms respectively.

[2]Each player has a game ID, assigned by the online game system based on when he entered the game room.

[3]*e.g.*, http://www.3j3f.com/how/

2.2 Attitude-Target Pairing

Then we associate each attitude word with its corresponding target. We remove interrogative and exclamatory sentences and only keep the sentences that include at least one attitude word from a player's statement during each round.

We develop a rule-based approach for attitude-target pairing: if there is at least one ID in the sentence, we associate all attitude words in that sentence with it. Otherwise, if "I" is the only subject or there are no subjects at all, we associate attitude words with the ID of the speaker. We reverse the polarity of an attitude word if it appears in a negation context.

Previous methods pair a target and an attitude word if they satisfy at least one dependency rule (*e.g.*, (Somasundaran and Wiebe, 2009)). We check the POS tag sequence between them. For each attitude-target pair, if there exists an attitude word, a belief-oriented verb such as "*think*", "*believe*", "*feel*", or more than two verbs in the sequence, we will discard this pair. The assumption is that POS tag sequences can be used to summarize dependency rules when statements are relatively short.

For those targets, the speaker didn't mention or there is no positive/negative attitude word used when they are mentioned, the attitude polarity score is set to 0. For instance, given Player 16's statement in Figure 2, its attitude tuple list is: [(16, 16, +1), (16, 11, -1), (16, 2, -1), (16, 1, 0), (16, 3, 0), . . . , (16, 15, 0)].

3 Clustering

Since the statements in conversations are relatively short and concise, it is difficult to identify which one is deceptive, even using deep linguistic features such as the language style.

In this section, we introduce a method to construct an attitude profile for each player and a signed network based on the attitude tuple list in Section 2, and combine them to analyze a dynamic network with discussants telling lies and truths.

3.1 Clustering based on Attitude Profile

We use a vector containing numerical values to represent each player's attitude toward identified targets in each round. The values correspond to the polarity scores in a player's attitude tuple list. For example, the polarity score of player 16's attitude toward target 11 is -1 as shown in Figure 2.

We call this vector as the discussant attitude profile (DAP) following (Abu-Jbara et al., 2012a).

Suppose there are n players who participate in a single game. Since a player's identity is not exposed to the public after his death[4], people can still analyze the identity of a "*dead*" player. Therefore, the number of possibly mentioned targets in each round equals to n. Given all the statements from m surviving players in a single round, each player's DAP has $n+1$ dimensions including his vote and thus we can have a $m \times (n+1)$ attitude matrix A where A_{ij} represents the attitude polarity of i toward j we got from Section 2. $A_{i(n+1)}$ represents i's vote.

In a certain round, given a set of m surviving players $X = \{x_1, x_2, \cdots, x_m\}$ to be clustered and their respective DAPs, we can modify the Euclidean metric to compute the differences in attitudes and get an $m \times m$ distance matrix M:

$$M_{ij} = \sqrt{\sum_{k=1}^{n}(A_{ik} - A_{jk})^2 + (2 - 2\delta_{A_{i(n+1)}, A_{j(n+1)}})^2} \quad (1)$$

The Kronecker delta function δ is:

$$\delta_{ij} = \begin{cases} 1 & i = j \\ 0 & i \neq j \end{cases} \quad (2)$$

We use this function to compare the votes of two players separately because a player's vote can be inconsistent with his previous statements. We assume that there is a larger distance between two players when they vote for different suspects.

A common assumption in previous research was that a member is more likely to show a positive attitude toward other members in the same group, and a negative attitude toward the opposing groups (Abu-Jbara et al., 2012a). However, a deceiver may pretend to be innocent by supporting those truth-tellers and attacking his teammates, whose identities have already been exposed. Therefore, it is not enough to judge the relationship between two players by simply measuring the distance between their DAPs.

In addition to comparing DAPs between players i and j, we also consider the attitudes of other players toward i and j, as well as their attitudes toward each other. We modify M_{ij} as follows and show it in Figure 3:

$$M'_{ij} = M_{ij} + \sqrt{\sum_{k=1}^{m}(A_{ki} - A_{kj})^2 + (h(A_{ij}) + h(A_{ji}))^2} \quad (3)$$

where the function h detects the negative attitudes. $h(x) = 0$ if $x \geq 0$ and $h(x) = -1$ otherwise.

We perform hierarchical clustering on the condensed distance matrix of M and use the complete linkage method to compute the distance between two clusters (Voorhees, 1986). We set the number of clusters as 3 since there are three natural groups in the game. We focus on separating deceivers (killers) from truth-tellers (citizens and detectives).

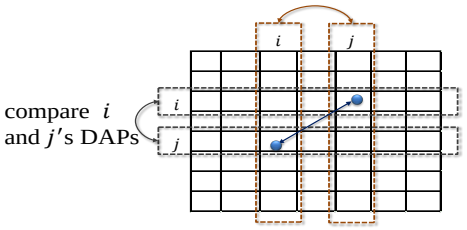

Figure 3: Computation of the distance between player i and j based on the attitude matrix.

3.2 Signed Network Partition

When we computed the distance between two players in Section 3.1, we did not consider the network structure among all the players. For example, if A supports C, B supports D and C and D dislike each other, A and B may belong to different groups. Thus, we propose to capture the interactions in the social network to further improve the attitude-profile-based clustering result.

We can easily convert the attitude matrix A into a signed network by adding a directed edge $i \rightarrow j$ between i and j if $A_{ij} \neq 0$. We denote a directed graph corresponding to a signed network as $G = (V, S, N, W)$, where V is the set of nodes, S is the set of positive edges, N is the set of negative edges and $W : (V \times V) \rightarrow \{-1, 1\}$ is a function that maps every directed edge to a value, $W(i, j) = A_{ij}$.

We use a greedy optimization algorithm (Doreian and Mrvar, 1996) to find partitions. A criterion function for an optimal partitioning procedure

[4]Each round, the player killed by killers and the player with the most votes are out.

is constructed such that positive links are dense within groups and negative links are dense between groups. For any potential partition $\mathbb{C}$, we seek to minimize the following error function:

$$E(\mathbb{C}) = \sum_{C \in \mathbb{C}} [(1-\gamma) \sum_{\substack{i \in C \\ j \notin C}} W(i,j) S_{i,j} - \gamma \sum_{i,j \in C} W(i,j) N_{i,j}] \tag{4}$$

where $\gamma \in [0,1]$ controls the balance of the penalty difference between putting a positive edge across and a negative edge within a group. We regard these two types of errors as equally important and set $\gamma = 0.5$ for our experiments.

Initially, we use the clustering result in Section 3.1 to partition nodes into three different groups and an error function, E, is evaluated for that cluster. Every cluster has a set of neighbor clusters in the cluster space. A neighbor cluster is obtained by moving a node from one group to another, or exchanging two nodes in two different groups. E is evaluated for all the neighbor clusters of the current cluster and the one with the lowest value is set as the new cluster. The algorithm is repeated until it finds a minimal solution[5]. We set the upper limit for the number of subgroups to 3.

3.3 Cluster Ensembles

The relationships between players are dynamic throughout the game. For example, a killer tends to hide his identity and pretends to be friendly to others at later stages in order to survive. Thus, it is insufficient to rely on a single round's discussion to cluster players. In addition, for each single round, we also need to combine the clustering results from the attitude profiles of the players and the signed network.

In a game with information gathered from up to r rounds, let $P = \{P_1, P_2, \cdots, P_r\}$ be the set of r clusterings (partitionings) based on attitude profiles and $P^{'} = \{P_1^{'}, P_2^{'}, \cdots, P_r^{'}\}$ be the set of r clusterings based on the signed network.

Using the co-occurrence relations between players, we can generate a $n \times n$ pairwise similarity matrix T based on the information of all r rounds:

$$T_{ij}^{r} = \frac{\lambda \cdot vote_{ij} + (1-\lambda) \cdot vote_{ij}^{'}}{r_{ij}} \tag{5}$$

[5]Since our graphs are small, we search through all partitions. We repeated 1000 times in our experiment.

where $vote_{ij}$, $vote_{ij}^{'}$ are the number of times that player i and j are assigned to the same cluster in P and $P^{'}$ respectively. r_{ij} denotes the number of rounds when both of them survived ($r_{ij} \leq r$). $T_{ij}^{r} \in [0,1]$. We assign a higher weight to the result of P_1 and set $\lambda = 2/3$ in our experiments.

Given the input in Figure 2, x_3 and x_4 are assigned to the same cluster in P_1 ($vote_{34} = 1$) and in $P_1^{'}$ ($vote_{34}^{'} = 1$) respectively as shown in Figure 4. x_3 and x_4 co-occurred in the first round ($r_{34} = 1$). $T_{34}^{1} = (2/3 \times 1 + 1/3 \times 1)/1 = 1$.

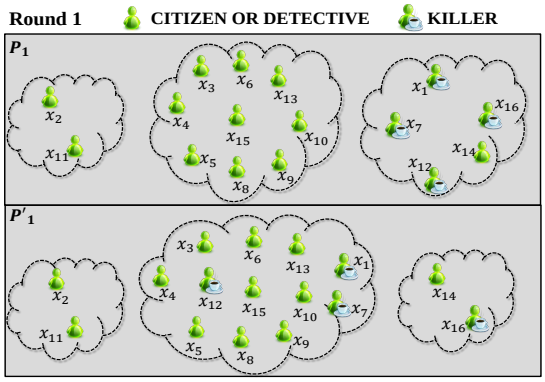

Figure 4: Example of cluster ensemble for a single round.

We apply hierarchical clustering (Voorhees, 1986) to the similarity matrix above to obtain the final global clustering results.

4 Experiments

4.1 Dataset Construction

We recorded 10 games from 3J3F[6], one of the most popular Chinese online killer game websites [7]. A screenshot of the game system interface is shown in Figure 5. There are 16 participating players per game: 4 detectives, 4 killers and 8 citizens. Each player occupies a position in ①. All the surviving players can express their attitudes via a voice channel using ②, while detectives and killers can also communicate with teammates in their respective private team channels ③ via texts. The system provides real-time updates on the game progress, voting results, and so on using the public channel ④. We manually transcribed speech and stored the text information in the public channel, which contains the voting and death information. The average game length

[6]http://www.3j3f.com

[7]All data sets and resources will be made available for research purposes upon the acceptance of the paper.

Game #	*Purity* (%)					*Entropy*				
	D	N	H	eD	$eD+N$	D	N	H	eD	$eD+N$
1	68.8	75.0	75.0	68.8	75.0	0.48	0.50	0.78	0.63	0.50
2	75.0	68.8	68.8	43.8	81.3	0.71	0.69	0.81	0.73	0.43
3	43.8	81.3	56.3	75.0	75.0	0.77	0.67	0.81	0.72	0.72
4	75.0	62.5	75.0	93.8	93.8	0.78	0.68	0.74	0.28	0.28
5	62.5	75.0	81.3	75.0	75.0	0.61	0.50	0.61	0.72	0.72
6	81.3	81.3	75.0	81.3	81.3	0.64	0.38	0.74	0.60	0.60
7	81.3	75.0	81.3	81.3	87.5	0.65	0.70	0.68	0.51	0.51
8	87.5	75.0	75.0	93.8	93.8	0.41	0.73	0.78	0.23	0.23
9	75.0	43.8	75.0	81.3	87.5	0.76	0.80	0.78	0.67	0.49
10	62.5	75.0	87.5	81.3	81.3	0.78	0.60	0.51	0.61	0.67
Average	71.3	71.3	75.0	77.5	**83.2**	0.66	0.62	0.72	0.57	**0.51**

Table 1: Results on subgroup detection. D refers to *DAPC*, N refers to *Network*, H refers to *Human Voting*, and eD refers to *extended DAPC*.

is about 76.3 minutes and there are on average 5 rounds and 411 sentences per game. Note that our method is language-independent and could easily be adapted to other languages.

Figure 5: Screenshot of the online killer game interface.

4.2 Evaluation Metrics

We use two metrics to evaluate the clustering accuracy: Purity and Entropy. Purity (Manning et al., 2008) is a metric in which each cluster is assigned to the class with the majority vote in the cluster, and then the accuracy of this assignment is measured by dividing the number of correctly assigned instances by the total number of instances N. More formally:

$$purity(\Omega, C) = \frac{1}{N}\sum_{k} max_j |w_k \cap c_j| \quad (6)$$

where $\Omega = \{w_1, w_2, \cdots, w_k\}$ is the set of clusters and $C = \{c_1, c_2, \cdots, c_j\}$ is the set of classes. w_k is interpreted as the set of instances in w_k and c_j is the set of instances in c_j. The purity increases as the quality of clustering improves.

Entropy (Steinbach et al., 2000) measures the uniformity of a cluster. The entropy for all clusters is defined by the weighted sum of the entropy of each cluster:

$$Entropy = -\sum_{j}^{} \frac{n_j}{n} \sum_{i}^{} P(i,j) \times log_2 P(i,j) \quad (7)$$

where $P(i,j)$ is the probability of finding an element from the category i in the cluster j, n_j is the number of items in cluster j and n is the total number of items in the distribution. The entropy decreases as the quality of clustering improves.

4.3 Overall Performance

We compare our approach with two state-of-the-art subgroup detection methods and human performance as follows:

1. DAPC: In Section 3.1, we introduced our implementation of the discussant attitude profile clustering (DAPC) method proposed in (Abu-Jbara et al., 2012a). In the original DAPC method, for each opinion target, there are 3 dimensions in the feature vector, corresponding to (1) the number of positive expressions, (2) negative expressions toward the target from the online posts and (3) the number of times the discussant mentioned the target. For our experiment, we only keep one dimension representing the discussant's attitude (positive, negative, neutral) toward the target since a discussant attitude remains the same in his statement within a single round.
2. Network: We also implemented the signed network partition method for subgroup detection proposed by (Hassan et al., 2012). To determine the number of subgroups t, we set an upper limit of $t = 3$ in order to minimize the optimization function.

3. Human_Voting: We also compare our methods with human voting results. There are two subgroups based on the voting results. The players with the highest votes each round belong to one subgroup and the rest of the players are in the other subgroup.

Table 1 shows the overall performance of various methods on subgroup detection and Figure 6 depicts the average performance. We can see that our method significantly outperforms two baseline methods and human voting. The human performance is not satisfying, which indicates it's very challenging even for a human to identify a deceiver whose deceptive statement is mixed with plenty of truthful opinions (Xu and Zhao, 2012).

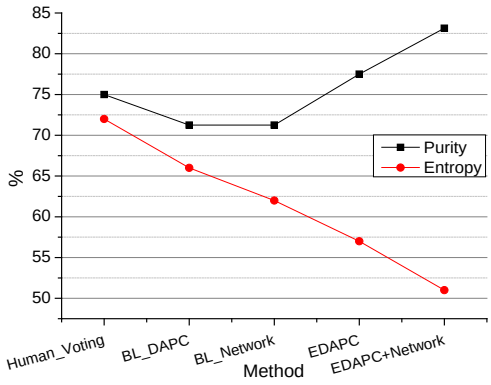

Figure 6: An overview of the average performance of all the methods.

By extending the DAPC method (EDPAC), we can estimate the distance between two players more accurately by considering the attitudes of other players toward them and their attitudes toward each other. Given the log in Figure 2 as input, players 5 (detective) and 7 (killer) are clustered into one group when DAPC is applied since they don't have conflicting views on the identities of other players. However, 5 voted for 7 and is supported by more players compared with 7, which indicates that they are less likely to be teammates. We can successfully separate them after re-computing the distance between them.

Adding network information provided 5.7% further gain in Purity. In some cases, the performance remains the same when EDAPC clustering result is already optimal with the minimum value of the criterion function.

4.4 Dynamic Subgroup Detection

As shown in Figure 7, the performance of our approach improves as the game proceeds. Players seldom maintain their opinions throughout a game. Figure 2 shows that most killers (16,1,12) insisted that citizen 11 should be a killer except 7. As a response to the group pressure (Asch, 1951), 7 changed his opinion and stated that 11 could be a killer in the following round.

In reality, a discussant who participates in an online discussion tends to change his opinions about a target as he learns more information, which shows both the necessity and importance of the dynamic detection of subgroups. Our method can be applied to detect subgroups dynamically by grouping posts into multiple discussion "*rounds*" based on their timestamps.

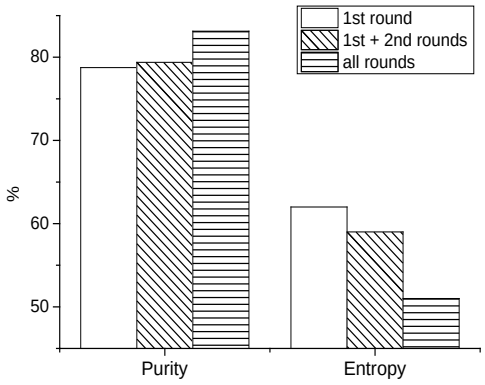

Figure 7: Average performance based on different rounds.

5 Related Work

5.1 Opinion Analysis

Our work on mining a player's attitude toward other players is related to opinion mining. Attitudes and opinions are related and can be regarded as the same in our task. Compared with the previous work (*e.g.*,(Qiu et al., 2011; Kim and Hovy, 2006)), the opinion words and targets in our task are relatively easier to recognize due to the simplicity of statements. Some recent work (*e.g.*, (Somasundaran and Wiebe, 2009; Abu-Jbara et al., 2012a)) developed syntactic rules to pair an opinion word and a target if they satisfy at least one specific dependency rule. We use POS tag sequences to efficiently help us filter out irrelevant pairs.

5.2 Deception Detection

Most of the previous computational work for deception detection used supervised/semi-supervised classification methods (Li et al., 2013b). Besides lexical and syntactical features (Ott et al., 2011; Feng et al., 2012; Yancheva and Rudzicz, 2013), Feng and Hirst (2013) proposed using profile compatibility to distinguish fake and genuine reviews. Xu and Zhao (2012) used deep linguistic features such as text genre to detect deceptive opinion spams. Banerjee et al. (2014) used extended linguistic signals such as keystroke patterns. Li et al. (2013a) used topic models to detect the difference between deceptive and truthful topic-word distribution. Researchers have began to realize the importance of analyzing computer-mediated communication in deception detection. Zhou and Sung (2008) conducted an empirical study on deception cues using the killer game as a task scenario and obtained many interesting findings (*e.g.*, deceivers send fewer messages than truth-tellers).

Our work is most related to the work of Chittaranjan and Hung (2010) on detecting deceptive roles in the Werewolf Game which is another variant of the killer game. They created a Werewolf data set by audio-visual recording 8 games played by 2 groups of people face-to-face and extracted audio features and interaction features for their experiments. However, we should note that non face-to-face deception detection emphasizes verbal and linguistic cues over less controllable non-verbal communication cues (Walther, 1996).

5.3 Subgroup Detection

In online discussions, people usually split into subgroups based on various topics. The member of a subgroup is more likely to show positive attitude to the members of the same subgroup, and negative attitude to the members of opposing subgroups (Abu-Jbara et al., 2012a). Previous work also studied subgroup detection in social media sites. Abu-Jbara et al. (2012a) constructed a discussant attitude profile (DAP) for each discussant and then used clustering techniques to cluster their attitudes. Hassan et al. (2012; 2012b; 2013) proposed various methods to automatically construct a signed social network representation of discussions and then identify subgroups by partitioning their signed networks. Qiu et al. (2013) applied collaborative filtering through Probabilistic Matrix Factorization (PMF) to generalize and improve extracted opinion matrices.

An underlying assumption of the previous work was that a participant will not tell lies nor hide his own stance. Moreover, their work did not take into account that a person's attitude or stance will change as he learns more by reading the comments from others and acquiring more background knowledge (Bandura, 1971). Our contribution is that we extend the DAP method and combine it with the signed network partition in order to cluster the hidden group members. We also develop a novel cluster ensemble approach in order to analyze the dynamic network.

6 Conclusions and Future Work

Using the killer game as a case study, we present an effective clustering method to detect subgroups from dynamic conversations with lies and truths. This is the first work to utilize the dynamics of group conversations for deception detection. Experiments demonstrated that truth-tellers and deceptive groups are separable and the proposed method significantly outperforms baseline approaches and human voting.

Our work builds a pathway to future work in deception detection in content-rich dynamic environments such as electronic commerce and repeated interrogation which will require sophisticated content and network analysis. In real-life suspects may be interrogated about particular events on numerous occasions. Our method can potentially be modified to find criminals who act in groups based on their statements. Other applications of this research include law enforcement, financial fraud, fraudulent ad campaigns and social engineering.

This study focuses on analyzing the verbal content in conversations. It will be interesting to study non-verbal features such as blink rate, gaze aversion and pauses (Granhag and Strömwall, 2002) when people play this game face-to-face and combine the non-verbal and verbal features for deception detection. In addition, it is worth exploring the impact of cross-cultural analysis in detecting deception. When attempting to detect deceit in people of other ethnic origin than themselves, people perform even worse in terms of lie detection accuracy than when judging people of their own ethnic origin (Vrij, 2000). For the future work, we aim to use automatic prediction of deceivers to help truth-tellers win games more easily.

Acknowledgement

This work was supported by the U.S. DARPA DEFT Program No. FA8750-13-2-0041, ARL NS-CTA No. W911NF-09-2-0053, NSF Awards IIS-0953149 and IIS-1523198, AFRL DREAM project, gift awards from IBM, Google, Disney and Bosch. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation here on.

References

A. Abu-Jbara, M. Diab, P. Dasigi, and D. Radev. 2012a. Subgroup detection in ideological discussions. In *Proc. Annual Meeting of the Association for Computational Linguistics (ACL 2012)*.

A. Abu-Jbara, A. Hassan, and D. Radev. 2012b. Attitudeminer: mining attitude from online discussions. In *Proc. North American Chapter of the Association for Computational Linguistics - Human Language Technologies (NAACL HLT 2012)*.

A. Abu-Jbara, B. King, M. Diab, and D. Radev. 2013. Identifying opinion subgroups in arabic online discussions. In *Proc. Association for Computational Linguistics (ACL 2013)*.

S. Asch. 1951. Effects of group pressure upon the modification and distortion of judgments. *Groups, leadership, and men. S*.

A. Balahur, R. Steinberger, E. Goot, B. Pouliquen, and M. Kabadjov. 2009. Opinion mining on newspaper quotations. In *IEEE/WIC/ACM International Joint Conferences on Web Intelligence and Intelligent Agent Technologies (WI-IAT 2009)*.

A. Bandura. 1971. *Social Learning Theory*. General Learning Corporation.

R. Banerjee, S. Feng, J. Kang, and Y. Choi. 2014. Keystroke patterns as prosody in digital writings: A case study with deceptive reviews and essays. In *Proc. Empirical Methods on Natural Language Processing (EMNLP 2014)*.

D. Buller and J. Burgoon. 1996. Interpersonal deception theory. *Communication theory*.

David B Buller, Judee K Burgoon, JA Daly, and JM Wiemann. 1994. Deception: Strategic and nonstrategic communication. *Strategic interpersonal communication*.

G. Chittaranjan and H. Hung. 2010. Are you awerewolf? detecting deceptive roles and outcomes in a conversational role-playing game. In *IEEE International Conference on Acoustics Speech and Signal Processing (ICASSP 2010)*.

P. Doreian and A. Mrvar. 1996. A partitioning approach to structural balance. *Social networks*.

V. Feng and G. Hirst. 2013. Detecting deceptive opinions with profile compatibility. In *Proc. International Joint Conference on Natural Language Processing (IJCNLP 2013)*.

S. Feng, R. Banerjee, and Y. Choi. 2012. Syntactic stylometry for deception detection. In *Proc. Association for Computational Linguistics (ACL 2012)*.

N. E. Friedkin. 2010. The attitude-behavior linkage in behavioral cascades. *Social Psychology Quarterly*.

P. Granhag and L. Strömwall. 2002. Repeated interrogations: verbal and non-verbal cues to deception. *Applied Cognitive Psychology*.

A. Hassan, A. Abu-Jbara, and D. Radev. 2012. Detecting subgroups in online discussions by modeling positive and negative relations among participants. In *Proc. Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL 2012)*.

S. Kim and E. Hovy. 2006. Extracting opinions, opinion holders, and topics expressed in online news media text. In *Proc. ACL-COLING 2006 Workshop on Sentiment and Subjectivity in Text*.

J. Li, C. Cardie, and S. Li. 2013a. Topicspam: a topic-model based approach for spam detection. In *Proc. Association for Computational Linguistics (ACL 2013)*.

J. Li, M. Ott, and C. Cardie. 2013b. Identifying manipulated offerings on review portals. In *Proc. Empirical Methods on Natural Language Processing (EMNLP 2013)*.

C. Manning, P. Raghavan, and H. Schütze. 2008. *Introduction to information retrieval*. Cambridge university press Cambridge.

M. Ott, Y. Choi, C. Cardie, and J. Hancock. 2011. Finding deceptive opinion spam by any stretch of the imagination. In *Proc. Association for Computational Linguistics (ACL 2011)*.

G. Qiu, B. Liu, J. Bu, and C. Chen. 2011. Opinion word expansion and target extraction through double propagation. *Computational linguistics*.

M. Qiu, L. Yang, and J. Jiang. 2013. Mining user relations from online discussions using sentiment analysis and probabilistic matrix factorization. In *Proc. North American Chapter of the Association for Computational Linguistics - Human Language Technologies (NAACL HLT 2013)*.

S. Somasundaran and J. Wiebe. 2009. Recognizing stances in online debates. In *Proc. Joint Conference of the Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP*.

M. Steinbach, G. Karypis, V. Kumar, et al. 2000. A comparison of document clustering techniques. In *Proc. KDD 2000 workshop on text mining*.

E. Voorhees. 1986. Implementing agglomerative hierarchic clustering algorithms for use in document retrieval. *Information Processing & Management*.

A. Vrij, P. Granhag, and S. Porter. 2010. Pitfalls and opportunities in nonverbal and verbal lie detection. *Psychological Science in the Public Interest*.

A. Vrij. 2000. *Detecting lies and deceit: The psychology of lying and implications for professional practice*. Wiley.

J. Walther. 1996. Computer-mediated communication impersonal, interpersonal, and hyperpersonal interaction. *Communication research*.

Q. Xu and H. Zhao. 2012. Using deep linguistic features for finding deceptive opinion spam. In *Proc. International Conference on Computational Linguistics (COLING 2012)*.

M. Yancheva and F. Rudzicz. 2013. Automatic detection of deception in child-produced speech using syntactic complexity features. In *Proc. Association for Computational Linguistics (ACL 2013)*.

H. Zhang, H. Yu, D. Xiong, and Q. Liu. 2003. Hhmm-based chinese lexical analyzer ictclas. In *Proc. SIGHAN 2003 workshop on Chinese language processing*.

L. Zhou and Y. Sung. 2008. Cues to deception in online chinese groups. In *Proc. Hawaii International Conference on System Sciences (HICSS 2008)*.

L. Zhou, J Burgoon, J. Nunamaker, and D. Twitchell. 2004. Automating linguistics-based cues for detecting deception in text-based asynchronous computer-mediated communications. *Group decision and negotiation*.