Does Betrayal Hide in Linguistics?
Machine Learning Betrayal Detection in Diplomacy Game

Liu Yehao, Lin Yuanhui, Li Hangqian

Ecole Polytechnique Fédérale de Lausanne

1 Abstract

Friendship is fickle, and there may be clues hidden in the words from friendship to betrayal[2]. In our work, we mainly based on Vlad Niculae's research on linguistic harbingers of betrayal occurs in one online strategy game *Diplomacy*. We use machine learning methods like random forest, support vector machine, logistic regression to build a model that can judge whether a player will betray or not

2 Introduction

Diplomacy is a popular and fascinating strategy board game, which is shown in Fig.1. In this game, players play a role in one of the seven countries in World War I. The goal of this game is to control everywhere in the map. There are many seasons in a game. In each season, players can chat with each other to build friendship or move their armies to attack somewhere else on the map. There are thousands of different strategies in this game, and one of the most important decisions is to decide when to betray your allies.

Figure 1: Diplomacy game

Vlad did many analysis of this game by taking data of 500 games. He use different techniques to extract many features from players' conversation and try to find whether there are some signals for a player to betray. These features are like politeness of the words, number of sentence they said. He found there are some imbalances for certain features between a finally betray game and not betray game.

One simple and direct question come to our mind, can we predict whether a player will betray or not in a certain season by analysing the word he said? In our project, we decide to use the data set the same as the his data set. We notice that in the section four of the paper, the authors mention there are imbalance of these variables between final betray or not. Hence, it is worth trying to use these feature to predict if a player will betray or not in the next turn. We also noticed that in the section five, the author mentioned that there are imbalance of variables before the seasons leading up to betrayal. Therefore, we put the variables of four seasons before betray together and try to train model to predict whether a player will betray or not by machine learning techniques.

3 Data set

Our project use the data set from Vlad Niculae's work(2015)[2], which contains a collection of games between players in the Diplomacy games. A game consists of consecutive game seasons in which the two players can chat, support or attack each other. The reason to use this data set is we could compare our result with the original paper directly. If we use a different data set, we might have a high probability to have different results due to different pre-processing methods and different data set. It is more convincing to say our model is better if our method increases the accuracy when we use the same data set.

We get our training features in two ways. The first is to keep the following 11 variables before the last act of friendship in each game. The variables and descriptions are shown in Table 1.

Feature	Description
sent pos	Positive sentiment
sent neu	Neutral sentiment
sent neg	Negative sentiment
discourse comp	Discourse complexity
plan	Planning level
argu claim	Argumentation level of claim
argu premise	Argumentation level of premise
n request	Number of requests
politeness	Politeness
n words	Number of words
n sentences	Number of sentences

Table 1: Linguistic features

In this way, we get 719 imbalances of betrayers and victims without betrayal and 663 with betrayal. The second way is to obtain the 11 variables of the seasons before the last support season(included), which is a 4-season series of variables. If one of the players in the game does not speak at all for 4 seasons, then the game will be eliminated because there will be too many 0s in the data set.

4 Related work

In the previous work of Vlad, they explored linguistic features which are related to betrayal. They found that there are subtle but consistent patterns in how people communicate when they are going to betray. Friendships which will end in betrayal are imbalanced. For example, betrayers show more positive, plan less and be more polite than victims before the betrayal. Linguistic cues are used to predict whether the betrayal will happen in the end using logistic regression. The cross-validation accuracy is 0.57,which is higher than the human-predict result 0.52.

They also found that changes in balance could mark imminent betrayal. As the breakdown approaches, the betrayer becomes more positive but less polite. While the victims request more and show more politeness. They also train a classifier using the the features from the older seasons preceding the last friendly support to predict eventually betrayal. The best model achieves an F1 score of 0.31.

5 Methods

In this section, we are trying to predict whether a player will betray or not given the conversation they did. We in Vlad's work, he mentioned there is an imbalance of linguistic features between final betrayal or not. Hence, we try to use these features to predict whether a player will betray or not in the next turn. Vlad also mentioned that there is an imbalance of features before the seasons leading up to betrayal. Hence, we put the features of four seasons before the last friendly support together and try to train different classifiers to predict whether the game will end in betrayal or not.

5.1 Overall features prediction

5.1.1 Distribution change of imbalance

We plot the imbalances of the 11 features between the betrayer and victim before the last support into a histogram, which is shown in Fig.2, and find that imbalances peak moves significantly to the right, which means imbalance between the betrayer and victims become larger in the eventual betrayal games in requests, words, and sentences number than in non-betrayal games.

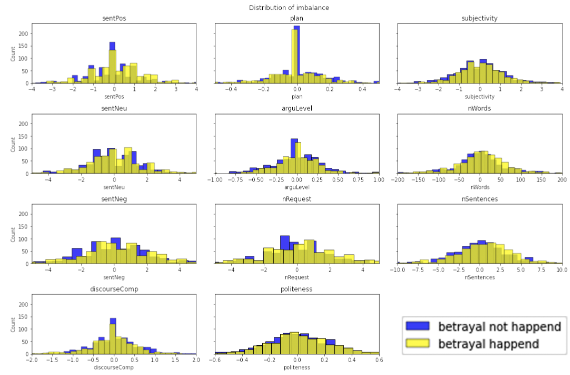

Figure 2: Distribution of imbalance in betrayal games and non-betrayal games

5.1.2 Pearson's correlation of features

For each feature in the data set, we compare its distribution in the treated group with its distribution in the control group and compute the Pearson correlation, which is shown in Fig.3. We find that sentiment negative has a high correlation with words number, sentence number, and request number. Sentiment positive shows slightly lower correlation with words, sentences, and requests number. Comparison is highly correlated with expansive. Also words number, sentences number, and request number are highly correlated with each other.

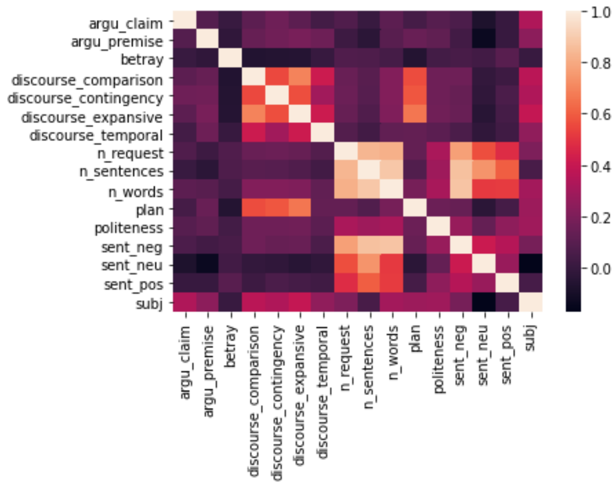

Figure 3: Heatmap of feature Pearson correlation

In our next work, we will use the highly correlated features for logistic regression analysis; meanwhile we will use cross-validation method.

5.1.3 Predictive power

We add the label betrayal to the data set, and mark it as 1 if betrayal occurs, and mark it as 0 if no betrayal occurs. we randomly split the data set into 0.90 training data set and 0.10 test set, and use logistic regression to train the classifier. The best accuracy is obtained by only keeping request number, words number, sentences number, and the interaction between Negative sentiment and

Sentences number features, which is 0.61. The Auc value is 0.59.

Then, we use 5-fold cross-validation to compute the accuracy. However, when we apply it on logistic regression, random forest and SVM, the highest accuracy we can get is only 0.54. So we are going to use the 4 seasons' time series to predict eventually betrayal.

We also use unsupervised learning to classify the data set. But when we use TSNE and PCA to reduce the 11-dimension data set to two dimensions, we find the betrayal and non-betrayal data highly overlap, which means we cannot effectively classify it into two sets. The result of k-means also shows that the data set cannot be effectively classified in this way, , which is shown in Fig.4.

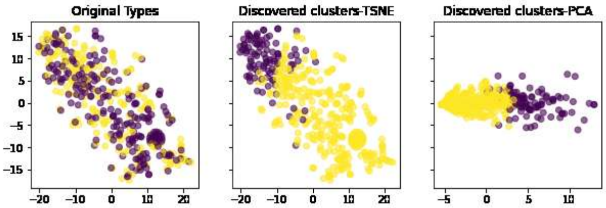

Figure 4: K-means result

5.2 Time series features prediction

Here we are going to present the second method we use. In this section, we combine the 14 variables of four seasons before betray together to get a 56 variables data set. Instead of treating each season individually, we treat each game individually and predict whether a player will betray in a certain season after given four seasons of conversation. Here we use some non-linear machine learning method as well. For example, we use support vector machine and random forest to fit the data set and predict whether they will betray or not.

5.2.1 Support vector classifier

At first we use support vector machine. we split 80 percent of data as training set and the rest 20 percent as testing set. We tried many different feature engineering but didn't find a good way to improve our accuracy. So we just do standardization for the training data set. We did a 5-fold cross-validation on our training set to find best hyperparameters, and get a 59.6 percent of accuracy on the cross validation. For the testing set, we get a 60 percent of accuracy on classifying them. Compared with random guessing, which follows distribution Bin(85,0.5). Use the command 1-pbinom(51,85,0.5) in r language we get 0.025. Our hypothesis is that our model is the same as random guessing. The p-value we just calculated is 0.025. Thus we reject the null hypothesis that our model is the same as random guessing and take the alternative hypothesis that our model is better than random guessing

5.2.2 Random Forest

For the random forest, split 80 percent of data as training set and the rest 20 percent as testing set. We did a 5-fold cross-validation on our training set to find best hyperparameters, and get a 59.1 percent of accuracy on the cross validation. For the testing set, we get a 60 percent of accuracy on classifying them. Compared with random guessing, which follows distribution Bin(85,0.5). Use the command 1-pbinom(51,85,0.5) in r language we get 0.025. Our hypothesis is that our model is the same as random guessing. The p-value we just calculated is 0.025. Thus we reject the null hypothesis that our model is the same as random guessing and take the alternative hypothesis that our model is better than random guessing.

We found that we can indeed find a strategy that performs better than random guessing on the probability that a player will betray.

6 Conclusions

We use features' imbalance based on all seasons before last friendly support and features data based on four seasons, which are three seasons before last friendly support and the last friendly support season, and found it can predict the betrayal significantly better than randomly guessing. Hence, we can develop a software to detect whether your friendly friend will betray you or not by inputting all the sentences he or she said.

In emotion detection, there may be some emotions that are not easy to predict, such as negative or Sarcasm Detection, which may have a negative impact on the true prediction of betrayal.[1]. Vrji(2004) reported that experienced professionals such as police officers have an average accuracy of 65% when asked to detect lies [3]. Compared with it, our model do not have any information about players' facial expression and body language but can perform a 60 percent of accuracy.

One of the challenging future work is to increase the accuracy of predicting. Maybe there are different kinds of way to process these verbal data so that the accuracy of lying detection can be increased.

References

[1] T. Jain, N. Agrawal, G. Goyal, and N. Aggrawal. Sarcasm detection of tweets: A comparative study. In *2017 Tenth International Conference on Contemporary Computing (IC3)*, pages 1–6, Los Alamitos, CA, USA, aug 2017. IEEE Computer Society.

[2] Vlad Niculae, Srijan Kumar, Jordan Boyd-Graber, and Cristian Danescu-Niculescu-Mizil. Linguistic harbingers of betrayal: A case study on an online strategy game. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1650–1659, Beijing, China, July 2015. Association for Computational Linguistics.

[3] Aldert Vrij. Why professionals fail to catch liars and how they can improve. *Legal and criminological psychology*, 9(2):159–181, 2004.