Predicting whether and when a betrayal will occur based on linguistic features

Ke Wang, Ruizhi Luo, Shanci Li
CS-401, EPFL, Switzerland

***Abstract*—One thing we all do not want to admit is that, even in very close relationships, betrayal still happen occasionally. Therefore, predicting if betrayal is near us is an important task. In our work, we focus on linguistic cues in messages between players in an online game Diplomacy. The game data provides massive information on the linguistic cues which could possibly be related to potential betrayals.**

In our work, the difference of the linguistic features of games ending with betrayal and friendships are captured in game-level, and utilizing these features, a model to classify whether a game would end up in betrayal is built. Knowing that betrayal is coming is not enough, possibly an equally important question is when will betrayal take place, so that precautions can be taken. In our work, another model to discriminate the last friendly season before betrayal from all the older seasons is then established. With these two models combined, we naturally came to the solution of the problem: Whether and when will I be betrayed?

I. Introduction

The study of interpersonal relations is gaining increasing attention from computational social scientists, despite the complexity of social interactions. One challenge for researchers is that hardly can they have the access to the whole data defining the relationship of friendships and betrayals.

In the paper of Vlad Niculae et al. [1], the problem in shortage of full data is addressed well by utilizing the data of Diplomacy, a war-themed strategy game where friendships and betrayals are orchestrated primarily through language. The data of this game is well-explored by Ref. [1] to study the the link between linguistic cues and potential betrayal.

A typical game in Diplomacy contains many seasons, in which players send messages to each other and have either friendly or hostile interactions. This is a good dataset to study linguistic cues linked to betrayal, as betrayals happen regularly in the game. The linguistic features of the message between the players are captured, and then explored in Ref. [1] to see their predictive power of whether friendships will end in betrayal and, if so, when the betrayal will happen.

However, we think that there are some assumptions in Ref. [1] which make its models impractical and illogical. The most important one is that the paper considers seasons as classification instances, rather than games. This can lead to confusion, when the prediction results of the seasons from the same game do not match with each other. This problem also shows up while predicting when the betrayal happens: If several seasons from the same game are all predicted to be immediately before betrayal, it can be quiet confusing.

Therefore, in our work, we consider games as classification instances to classify whether they would end with betrayal or friendships, and when discriminating the last friendly season, we would make sure that in each game, there would only be season which is immediately before betrayal.

A practical problem is that, given the data of a game, whether this game would end up in betrayal, and if yes, in which season would the betrayal happen? Although this article proposed methods to solve different parts of this problem (whether and when), their results can not be connected to address this particular problem. This is precisely the problem we wish to solve in our paper, and to achieve this, our pipeline approach is sketched in Figure 1. As shown in the Figure, we would first predict whether the game would end up in betrayal (through Model I), and if yes, we would further make predictions on which season would betrayal happen (through Model II), thereby solving this problem.

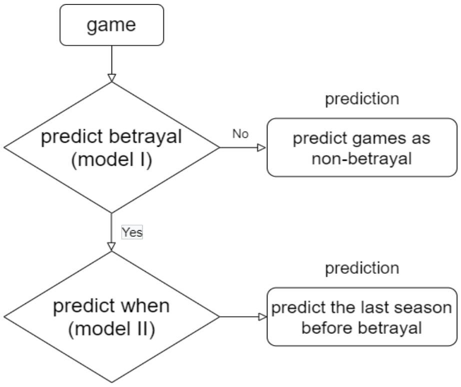
game
predict betrayal (model I)
No
prediction
predict games as non-betrayal
Yes
predict when (model II)
prediction
predict the last season before betrayal

Fig. 1. The pipeline for our method

The structure of this paper is as follows: In Section 2 we will briefly describe how we processed the data based on the work of Ref. [1]. Then, in Section 3 we will introduce our proposed Model I to predict whether a game would end up in betrayal, and in Section 4 we will describe Model II to predict when the betrayal might happen. In Section 5, we will show that the results of the models in Section 3 and Section 4 could be well combined to simultaneously predict whether and when a betrayal might happen.

II. Data Processing

We will use the same data set as Ref. [1] because it is difficult to obtain new datasets for this game. On top of their work, we did some preprocessing to ensure the effectiveness of data.

For each game, we keep only seasons when both players send messages to each other, so that we can compute the imbalance of betrayers and victims.

Furthermore, we discard meaningless messages in which the number of sentences is zero.

III. Model I: Predicting whether the game would end with betrayal

In Ref. [1], the article applies a binary classification in which it considers the seasons coming from games ending with betrayal or friendships as classification instances.

However, choosing seasons as classification instances can lead to confusion. If the prediction results on several seasons from the same game conflict with each other (some are predicted to be betrayal seasons and others are not), should we classify this game into betrayal or non-Betrayal?

Hence, rather than seasons, we consider games as classification instances to predict whether they would end with betrayal or friendship. After data pre-processing, we got 193 games ending with betrayal and 189 games ending with friendship. As for features, we average over the linguistic cues in Table 2 of Ref. [1] of each season in the game. From Figure 2, we can see an obvious difference in these normalized features between games ending with betrayal and friendship, suggesting a strong link between linguistic cues and potential betrayal in game-level.

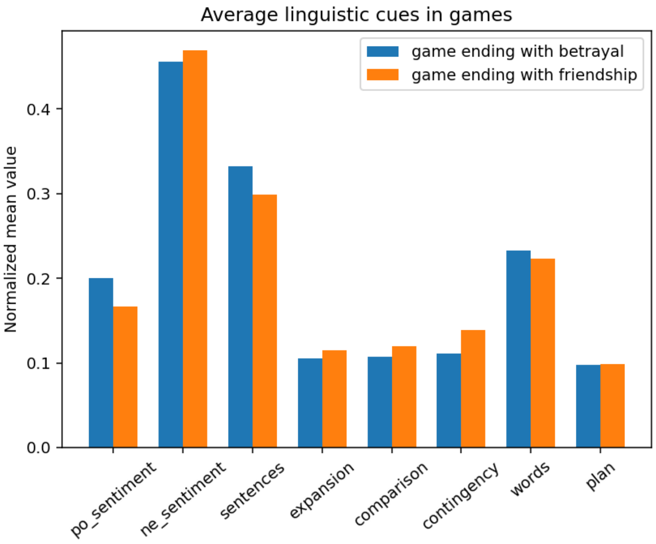

Fig. 2. The comparison of features of games ending with betrayal and friendships

We label each game with a game index, which is helpful when we group season instances from the same game in Model II (will be explained in Section 4). We randomly shuffled the game indexes and split the data into training set and test set according to the game indexes.

In the training process, we adopted and compared three different algorithms, namely linear regression, gradient boost regressor and neural network. After 5-fold cross validation for 100 times for each of these three models, the result suggests that Gradient Boosting Regressor outperforms the other two algorithms (see the result in Figure 3). It achieves a mean cross-validation accuracy of 57.9% and a mean Matthews correlation co-efficient of 0.164, both slightly better than that the result in Ref. [1], which verifies that the linguistic features indicating potential betrayal can be better captured at the game level than season level.

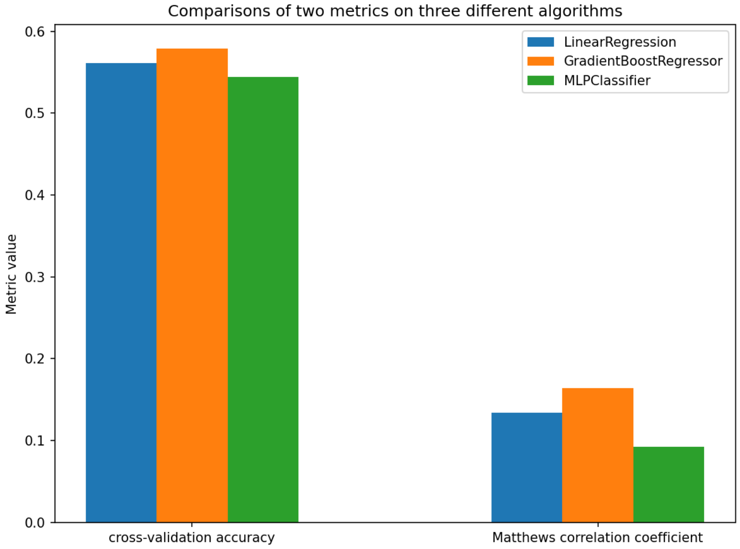

Fig. 3. The comparison of three algorithms in terms of the cross-validation results

IV. Model II : Predicting when would betrayal happen

In section 5 of Ref. [1], the authors trained a classifier to discriminate between the season preceding the last friendly interaction and all the older seasons. However, the author again took the seasons as classification instances, completely ignoring their game-level relevance. So, the problem emerges again, if several seasons of the same game are predicted to be the last friendly season (obviously in a game, there is only one last friendly season), which season should be considered as the ‘real’ last friendly season?

Hence, in this section, this problem is addressed in our approach. We also consider seasons as classification instances, but in the mean time we also take the game-level information of each season into account. Iterating over the games ending with betrayal, we obtain the features described in Table 3 of Ref. [1] for each season. Then we consider only the games in which the friendships last for more than 2 seasons, and finally we got 641 seasons, with 23.6% of them being the last seasons before betrayal. The index of the game which these seasons belong to is also recorded, to later group the seasons which belong to the same game. The seasons right before betrayal are labeled ‘1’ and all the other older seasons are labeled ‘0’.

In Model I, the training set and test set are split according to the randomly shuffled game indexes. Here we use these indexes again, so that the games are split to the same set as Model I. Besides, splitting according to game indexes guarantee that all the seasons from the same game are split to the same set.

In the training process, Gradient Boosting Regressor again outperforms the other two algorithms mentioned in the previous chapter. The Gradient Boosting Regressor takes the features of the input season as input, and outputs the probability of this season being the last season before betrayal. To make sure that in each game, only one season is predicted to be the last season before betrayal, we group the seasons belonging to same game, and label the season with the maximum probability as '1', and label all the other seasons in this game to be '0'. Generating the predictions in this way is not only more logical than making predictions on individual seasons, but also achieves a higher F1 score. The prediction using our approach achieves a F1 score of 0.42, in contrast to 0.32, which is obtained by making individual predictions on each season.

V. Connecting Model I and Model II and Generate Final Result

Now we have trained Model I and Model II respectively, it is time to combine their prediction results on the test set to solve the problem we described in Figure 1. The game index stored in the prediction result in each model allows us to find these two models' prediction results on the each game. Given the data of a game in the test set, if this game is predicted to be non-betrayal by Model I, we stop here and do not seek the result of Model II of this game. If it is predicted to be betrayal by Model I and it is a true positive, we turn to the prediction of Model II, and record which season is the predicted last season before betrayal.

Eventually, in Table I we present several examples of our final prediction result.

TABLE I
Final Prediction Result

game index	betrayal label	predicted label	season before betrayal	predicted season before betrayal
9	1	0	2	Predicted to be non-betrayal
27	0	1	No betrayal	Not applicable
66	1	1	7	7
100	1	1	5	5
434	1	1	4	3

In Table I, the column 'game index' refers to the index of the game, column 'betrayal label' refers to the true label of whether the game eventually ends with betrayal, while the column 'predicted label' refers to the prediction of Model I on whether the game ends with betrayal. The column 'season before betrayal' indicates in which season did the betrayal happen (if the game actually ends with friendship, it is marked with 'No betrayal'). Finally the column 'predicted season before betrayal' records the prediction of Model II on which season the betrayal happened. If a betrayal game is miss-classified to be non-betrayal, it is marked as 'Predicted to be non-betrayal', and if a non-betrayal game is miss-classified to be betrayal, it is marked as 'Not applicable'.

VI. Summary

While the predictive power of linguistic cue predicting a forthcoming betrayal has been illustrated in Ref. [1], this article fails to consider the data from a game as a whole, making the discoveries less practical. In our work, we consider the classification instances at game level instead of season level, and trained two different models to predict whether the game would end up in betrayal and when the betrayal might take place. Importantly, these models can be integrated to a pipeline, allowing us to solve this practical question 'Will I be betrayed? If so, when will I be betrayed?'

There still remains future work to be done: Instead of 'predicting' on the existing dialogues, the ultimate goal for us is to design a real-time prediction system, which allows players in Diplomacy to see the how far away betrayal is near them while playing the game.

References

[1] V. Niculae, S. Kumar, J. Boyd-Graber, and C. Danescu-Niculescu-Mizil, "Linguistic harbingers of betrayal: A case study on an online strategy game," *arXiv preprint arXiv:1506.04744*, 2015.