Extension of Linguistic Harbingers of Betrayal

CS-401 Applied Data Analysis

Younes Belkada, Irina Madalina Bejan, Maxime Emschwiller

December 18, 2020

Abstract

Following our replication of the work done by Niculae et al [1] in his study **Linguistic Harbingers of Betrayal: A Case Study on an Online Strategy Game**, we explore further if imminent betrayal prediction can be improved, what features in a dyadic relationship are more indicative of betrayal or if words used play an important role to signal betrayal, through an exploratory analysis.

1 Introduction

The original paper [1] explains how the messages exchanged between every two players of an online strategy game, Diplomacy, contain subtle signs of imminent betrayal, which victim is not able to seize. Using the game data is a novel approach, giving access to more insights into the act of betrayal, where no consistent work or datasets were considered before. The researchers analyze different particularities of the exchanged messages, such as politeness and sentiment, ending on the possibility to train a model that could predict an imminent betrayal better than a human player. Relying on this result, we will explore the possibilities of improvement of the prediction by using different well known models of machine learning. Beside this objective of pure optimization, we will explore the possibilities of prediction in an arbitrary number of seasons ahead of the betrayal and attempt to understand what other features can be indicative of betrayal in the socially complex phenomena of betrayal. More explicitly, our study tries to find an answer for the following questions:

- Can we predict imminent betrayal better?
- What is the probability of getting betrayed as a player ?
- How early (how many seasons before the betrayal happens) can we predict the incoming betrayal?
- What features are more indicative of betrayal? Can the semantic of words help to foretell betrayal?

2 Dataset

Given that the dataset provided in the cited paper consists of high-level features extracted from messages, our freedom of extension becomes quite limited. We analyzed the possibility of other datasets that can complement our goal, but were not able to find original raw data or other sources containing betrayal clues. We found a project based on the Diplomacy game, consisting of 156k games and 13M messages, but we were not granted access to it. Thus, we have decided to work on the data provided by the authors which will give us the possibility to compare our methods with the authors' results.

2.1 Data wrangling

To predict the imminent betrayal in a comparable manner to the authors', we filter the instances similarly, by selecting the entries' seasons that happen before the last support action from any of the players. Furthermore, we consider only the cases where both the potential victim and betrayer have sent at least one message. We considered to use most features extracted in the dataset, as sentiment (positive, neutral, negative), number of requests, number of words, politeness score, number of sentences and the number of seasons before the potential betrayal.

To improve the prediction, we consider new features: the length of a friendship (as the difference between the first support action and last support action), the difference in the politeness score between betrayer and victim and the difference in politeness between current and previous season. We standardize features values using the mean of the training data. Out of 1895 seasons that end in betrayal, there are 343 seasons take place one season before betrayal and we will consider these seasons further to predict imminent betrayal.

3 Predicting betrayal

Our first goal is to improve the prediction of imminent betrayal, by discriminating between the season before betrayal and the other seasons. Our baseline model is the one employed by authors.

3.1 Metrics

For a fair comparison to the baseline, which obtains a *F1* score of 0.31 and a Matthews correlation coefficient of 0.17, we employ the same metrics :

$$MMC = \frac{TN * TP - FP * FN}{\sqrt{(TN + FN) * (FP + TP) * (TN + FP) * (FN + TP)}}$$

$$F1 = \frac{2 * Precision * Recall}{Precision + Recall}$$

$$Precision = \frac{TP}{TP + FP}, Recall = \frac{TP}{TP + FN}$$

With TP, FP, TN, FN corresponding respectively to the True Positives, False Positives, True Negatives and False Negatives.

3.2 Imbalance

As explained above, only 18% percentage of seasons are immediately before betrayal. As models tended to overfit on the majority class, we did oversampling of the positive class to have balanced samples extracted through bootstrapping. Moreover, we used the weights derived from the data distribution to increase the cost of predicting wrong the positive class.

3.3 Logistic regression

We trained first Logistic Regression to predict imminent betrayal. For improvement the results, we did a grid search on combinations of input features and considered the best one, obtaining very good results when applying bootstrapping..

3.4 Decision trees

As the models aggregating decision trees are among the most used and effective and were not used in the study, we use them to improve the model's scores. We explored using both Random Forest and Boosted Trees. We suspect that the lack of data lead to prevent the usefulness of deep trees and we noticed that using a small depth (4) and a big number of estimators (1000) for random forest worked better, while the ability of boosting helped get comparable results with a higher depth (100) and 300 estimators on XGBoost.[2]

3.5 Feed forward neural network

We attempt to train a feed forward neural network, consisting of two fully connected layers of 512 and 256 hidden units, with a tanh activation of the first layer and a sigmoid one to output the probability. We apply dropout [3] with a factor of 0.4 and L2 regularization to tackle overfitting. We used binary cross-entropy as the loss function, a learning rate of 0.0001 and the Adam [4] optimizer. We would train until the loss in the validation data starts to increase, which happens quite fast, given our oversized architecture compared to inputs.

3.6 Multimodal neural network

We hypothesize the semantic value carried by the words could help predict betrayal. While we do not have the original raw messages, we combine the frequent words and lexicon words to have a set words for each season.We remove all words smaller than 3 chars, that are not understandable, contractions and common (stop) words. For representation, we employ the lowest dimension word embedding of GloVe [5] (25 dim) and use a maximum of 200 words per season.

The model consists of two models combined by concatenating their hidden representation [6], which is a common method in deep learning for multimodal inputs. For the numerical values, a feed forward network is used, while for embeddings, we use a neural network based on long-term short memory units [7]. We used the LSTM units because of its proven ability to learn temporal dependencies in data [8]. Even if we don't have words in order, it performs better than a dense layer, retaining more semantic value of the messages. The first model uses 64 hidden units with tanh activation and the latter, 64 LSTM units, both using ridge regularization. The fusioned model is trained jointly, using a batch of 32, Adam [4] optimizer, binary cross entropy for the cost function, a learning rate of 1e-5 and dropout [3] before the sigmoid activation to prevent overfitting. We also explore the ability to predict of a model that uses only the embeddings and no numerical value.

4 Results

In order to evaluate our methods, we split the data into training and testing. Since our study is constrained by its small number of instances, we apply bootstrapping and we repeat our training procedure for a number of 20 iterations by randomly taking 90% of the dataset for training and the remaining data for testing. The results obtained are presented in Table 1 and the scores considered are averaged from all iterations.

Model	F1	MMC
Authors Baseline	0.31	0.17
Logistic Reggression	0.532	0.351
Random Forest	**0.561**	**0.490**
XGBoost	0.502	0.434
Feed forward neural network	0.454	0.287
Multimodal neural network	0.395	0.204

Table 1: Results of predicting imminent betrayal

5 Interpretation of results

Given random forest give us the best prediction, we look into the feature importance to understand **why**. As we can see in Figure 1, the length of friendship is the most indicative of whether betrayal will happen. A strong friendship may hardly last to a betrayal, while on the other hand a weak and short friendship may easily last to a betrayal. Furthermore, the difference in politeness between seasons is more important than the one between players. Our results are worse when using the multimodal model, showing that frequent words alone do not have a consistent contribution to the output. However, given our very little dataset, it is highly probable that even with a low dimension embedding, the model is too complex and overfits.

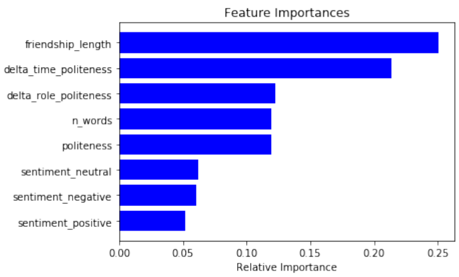

Figure 1: Importance of features in Random Forest model

Based on the better results we obtained, we explore further if we can foretell an incoming betrayal with confidence much early than in the season immediately before the imminent betrayal. We retrain the logistic regression and random forests models on the same task, but we try to predict whether a betrayal will happen in exactly 2,3 or 4 seasons from the current season. We therefore remove the seasons that would predict outside of our knowledge (i.e season before betrayal for a prediction window of 2) and drop the friendship feature. The results are much higher when predicting further into the future, with the highest score for 3 seasons before betrayal - as we can predict with a bootstrapped F1-Score of 0.3 and MMC score of 0.11 the incoming betrayal, considering a 95% confidence interval. This is a surprising result, that could potentially signal that betrayers show subtle signs of future betrayal much earlier.

6 Helping players

We can think of a deployment of our approach for detecting imminent betrayal as follows:

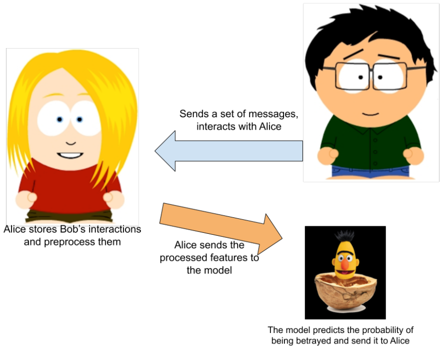
Sends a set of messages, interacts with Alice
Alice stores Bob's interactions and preprocess them
Alice sends the processed features to the model
The model predicts the probability of being betrayed and send it to Alice

Figure 2: A possible application of our method

During the playing season, two players Alice and Bob who are friends interact by sending messages to each other. At the end of the season, if Alice has a doubts, she could use messages from Bob, extract the features of them as described in the paper[1] and use our model to predict an imminent betrayal. Our work would be close to an AI-powered decision making by warning the potential victim for a incoming betrayal.

7 Conclusion

The results we obtained expand the insights gathered by the authors of Linguistic Harbingers of Betrayal [1], showing that betrayal can be predicted even more accurate through machine learning for the Diplomacy game. We found that the length of friendship is a great contributor to the prediction which reinforces our common understanding that longer friendships are harder to break. Yet, the ability to foretell betrayal within a much longer window size suggests that players might employ long-term strategies in the game.

References

[1] Vlad Niculae, Srijan Kumar, Jordan Boyd-Graber, and Cristian Danescu-Niculescu-Mizil. Linguistic harbingers of betrayal: A case study on an online strategy game, 2015.

[2] Tianqi Chen and Carlos Guestrin. Xgboost. *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, Aug 2016.

[3] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: A simple way to prevent neural networks from overfitting. *J. Mach. Learn. Res.*, 15(1):1929–1958, January 2014.

[4] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization, 2017.

[5] Jeffrey Pennington, Richard Socher, and Christopher D. Manning. Glove: Global vectors for word representation. In *Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, 2014.

[6] Kuan Liu, Yanen Li, Ning Xu, and Prem Natarajan. Learn to combine modalities in multimodal deep learning, 2018.

[7] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9:1735–80, 12 1997.

[8] Jeffrey L. Elman. Finding structure in time. *Cognitive Science*, 14(2):179 – 211, 1990.