Dota Chat Analysis

Sara Stanić
i6141502

Siemen Geurts
i6166476

***K*eywords** Dota 2 · Natural Language Processing · Sentiment Analysis · Match Outcome Prediction · Machine Learning · Slang Translation

1 Introduction

Video games have been a source of amusement, learning and challenge for youth to create greater winning strategies. However, that is not their sole purpose anymore. With rising technology and online play, most online games provide a chat system for players to communicate strategies, compliments or other important information about current game events. It is known that as time progresses so does the language, with each generation adding their own slang and expressions to communicate with their peers. Hence, chatting during a game has become more common and necessary, but also confusing for outsiders. Sentences like "GG" and "WP" are abbreviations usually used by players praising each other's game skills while "FU" has a completely opposite meaning. This gives us insight about the game as it progresses. People's reactions during a win and loss are challenging to analyze as some of the expressions cannot be found in dictionaries and cannot be studied by sentiment lexicons. Moreover, the use of irony and sarcasm is not rare during game play which makes it complex for a model to assess sentiment. In this work, we explore the connection between a game result and the player's chat messages, trying to track the events going on in the game based on communication.

2 Prior Literature

Match outcome prediction is the main task to be completed, and sentiment analysis plays a large role in evaluating a players' state throughout the game. Though sentiment analysis is no new concept, its use in video game chat systems is limited, due to a large amount of slang and game-specific jargon used to describe emotion or sentiment. An extensive slang-to-sentiment analysis has been performed before [4], saved in a so-called 'sentiment dictionary' (henceforth called SD), containing a sentiment score ranging from -2 to 2 for negative to positive words respectively. This dictionary was built from several sources, namely Urban Dictionary and Twitter, and will be further discussed in Section 3: Data.

Furthermore, some game-specific and extensive data-analysis regarding cyberbullying in gaming [3] shows promise in deriving connotation from chat messages. Here, the distinction of positive to negative is based on several features such as racist comments and targeted attacks, though source code is only available in C# and SQL, so incorporation into this paper will stick purely to conceptual use and as source of inspiration due to time constraints.

In hindsight, the SD was not used, as its implementation was deemed unnecessary and without any validation, sentiment analysis is hard to interpret.

3 Data

The dataset used in this paper (based on [2] and [1]) presents several tables regarding player's performance and behaviour (a total of 50,000 matches). As the data source is from an online game 'Dota 2', there is also a lot of match data regarding items bought, abilities cast, and more. However, for the purpose of this paper, these values are ignored as we are purely interested in chat analysis and its correlation with win rate. As such, several tables within the dataset are merged together to form a coherent and useful table. This table includes a match id, the team that won, a set of messages sent by the first team, and a set of messages sent by the second team. It is possible, though, that some additional feature detection such as those presented in [3] and dropping of words may be required in order to get sensible results. The data must also be cleaned before being put through the model. We start by transforming every abbreviation and emoticon into an English word or set of words. For this first task, a website was scrapped (netlingo.com) which contained a list of abbreviations with their meaning. Lastly, every form of 'hahaha' was transformed to 'lol' to prevent too many different types to appear in the data. Next, the sentences had to be transformed in such a way that the model could use it as input. This means transforming the sentences sent by each players into some numeric value. We do so, through a word2vec modeled trained on the dataset. We did not make use of pre-trained word2vec models, due to the lack of slang words and stop words.

Both of these are quite prevalent in online speech, so not having any training on these may have a negative impact on the power of the model. Each word is transformed using the word2vec model, and then every sentence is computed by averaging the total word-vectors in it. Next, every set of teams' messages is turned into a vector by averaging every sentence sentence-vector within it. Lastly, for each match, both teams message-vectors are combined and averaged to get one final vector which will be used as input for the model. There is some degeneracy in the data, due to Russian. This has been left in the data for the time being, as only later was discovered Python implicitly converts utf-8 format to ASCII at some point. This causes many questions marks (for unknown characters) to appear, which would indicate a question.

4 Model

As mentioned, there are two main sub-tasks presented in this paper, though only one will require the building of a model, namely match outcome prediction. Sentiment analysis is not a feasible task to tackle, considering there are no given labels. A way to avert this is by individually marking sentences as negative or positive (ranging from -2 to 2 respectively), though this is not possible due to time constraints. Therefore the results presented by this method will not be the main focus of this paper, but rather the second task: match outcome prediction. This model has as an input the set of messages sent by all players on each team per match, and outputs a single value (0 or 1), which corresponds to losing and winning said game respectively. Here, several approaches will be applied, namely Support Vector Machine, Logistic Regression and XGBoost. From these, we can determine whether chat messages are somehow correlated with a likelihood of winning.

5 Experiments

The experiments to be conducted are testing each model's accuracy and area under ROC-curve. These will give not only a good indication of which model might be most suited for the problem at hand, but also provide an insight into whether chat is in any way correlated with win rate. If time allows it, parameter tweaking and feature addition can be investigated. Features such as number of times 'good game' is said might indicate a positive win rate. Winning players tend to be more likely to say a game was good rather than not. Also, 'ez' may provide a good insight, as most players who say this may have an easy time during the match, indicating having the upper hand. However, features like these are to some degree already present in the word2vec model presented, due to the averaging of the vectors which means high numbers of 'good game' will move the average vector into that direction.

6 Results

After running initial experiments, we found that Logistic Regression performs worst of the 3 with approximately 0.65 accuracy. SVM took quite a bit longer to train, and gave an accuracy of 0.67. Both inputs were the exact same here. Similarly, this data was input into an XGBoost algorithm. Here, it appeared that after 100 iterations, there was still some improvement over time, but seemed so little that no more iterations were performed. XGBoost is not known for suddenly increasing its power over a single iteration, so the next 100 will most likely have had minimal improvement. XGBoost produced a AUC-value of 0.704, which we consider to be quite high (more in Section 7). No further experiments were run to increase the accuracy of either of these model due to time constraints. An important note is the distribution of the data. For the training set, approximately 51% of the data was positive, indicating team 1 won. For the test set, approximately 52% of the data was positive.

7 Analysis

The first thing to note is the fact that each classifier is quite drastically above the 50% margin, indicating that it is far better than random guessing. This may also indicate that chat indeed does provide useful insight into whether a player won or lost, especially considering the fact that no additional features were included in the tests. XGBoost performed best out of all 3 classifiers, but it took 100 iterations (though it was considerably faster than SVM and only a slight bit slower than Logistic Regression), though more iterations may be subject to overfitting to the data. The word2vec model used seemed to be quite a success as well, as finding the most similar word to words such as 'great', 'good game' and 'laughing out loud' gave very semantically similar responses, some even if the term is used in a sarcastic tone. As such, we believe that the greatest issue with the current state of the data is the amount of noise (more on this in Section 8). With more features, such as making use of the SD mentioned in 2, we are quite confident that our models' accuracy will increase.

8 Conclusion

The results generated from the performed tests give a strong suggestion that players' current state of the game (more likely to win vs. more likely to lose) affects how they interact with their allies and enemies. The greatest reason for being unable to fully conclude this from our test, is the great number of uneliminated noise. A lot of Russian, concatenated slang and typos in slang go unnoticed by our simple cleaning of the data, and must have a negative effect on the accuracy of our models. Future work should try to improve the cleaning of the data and include some form of sentiment analysis if possible.

9 Appendix

The dataset used during for this paper can be downloaded at https://kaggle.com/devinanzelmo/dota-2-matches.

References

[1] Albert Cui, Howard Chung, and Nicholas Hanson-Holtry. "OpenDota - All Matches from March 2016 - Matches". In: (). URL: opendota.com.

[2] Albert Cui, Howard Chung, and Nicholas Hanson-Holtry. "OpenDota - All Matches from March 2016 - Player Matches". In: (). URL: opendota.com.

[3] Shane Murnion et al. "Machine learning and semantic analysis of in-game chat for cyber bullying". In: *Computers Security* 76 (Mar. 2018). DOI: 10.1016/j.cose.2018.02.016.

[4] Liang Wu, Fred Morstatter, and Huan Liu. "SlangSD: Building and Using a Sentiment Dictionary of Slang Words for Short-Text Sentiment Classification". In: *CoRR* abs/1168.1058 (2016). URL: http://arxiv.org/abs/1608.05129.