Predicting Dota 2 match outcomes based on hero dynamics, chat features, and early game metrics

Widjaja, Joseph
jewidjaj@ucsd.edu

Khorram, Michael
mrkhorra.@ucsd.edu

Sengupta, Neil
nesengup@ucsd.edu

Abstract

This paper analyses Dota 2 multiplayer team matchups and tries to evaluate what team characteristics and matchups between heroes in teams are essential to bring positive outcomes for a team in a match. This paper attempts to model team composition and counters in addition to taking into account chat logs and the impact of early game for win prediction. Evaluations are made as to what features improved accuracy and eventual progression to using other diverse features.

Introduction

Dota 2 is a free-to-play action real-time strategy game made by Valve Corporation. The monthly player base sometimes exceeds 13 million unique users during peak times. Many people play the game casually, but there is also a professional scene. The game's popularity spikes especially during The International, an annual Dota 2 championship hosted by Valve in Seattle. The International 2017 had a prize pool of over $24 million spread across 18 professional teams, where the first place team Liquid claimed over $10 million. Most of the prize money was crowd funded.

Each match consists of 10 players divided into two teams of five. One teams is called the Radiant while the other is called the Dire. Before starting a match, each player selects one of the 115 unique heroes to control in the game and do battle against players from the other team. Once one of the heroes is selected, there can only be one of those heroes in the game, meaning another player cannot select that hero. Each hero has unique characteristics and abilities that provide interesting dynamics depending on heroes on the same team and heroes on the opposing team. During the hero selection process before the game starts, players often try to select heroes with the thought of how that hero contributes to the team and effectiveness against the heroes chosen on the other team. During the game, each player only controls one of these heroes and a match ends when one of the team's base has been destroyed. When a game starts, players always start their hero at level 1. Every 30 seconds minions that are not player controlled mindlessly, called creeps, charge toward the opposing base. Throughout the game, players are able to earn experience points (xp) to level up, and gold by giving the killing blow to creeps. Hero skills can be learned every time a hero levels up and items that augment heroes by giving them more capabilities and unity can be purchased with gold.

In Dota 2, there are several different matchmaking types and modes that users are able to play. Ranked matchmaking allows participants to wager their matchmaking points that serve as a metric of skill for a more competitive environment. Winning increases matchmaking rating (MMR), while losing decreases MMR.

In this study, ranked matches will only be considered because players take the games more seriously in ranked and actually try to win. In addition, the game restricts newer players from playing ranked matchmaking, so it provides a better representation of players utilizing heroes to their full potential.

Dataset

We found a dataset on Kaggle taken from the

YASP (opendota.com) database of 50000 ranked Dota 2 matches from late 2015 to perform our study. This dataset performed replay analysis to extract extra fields not accessible straight from the Valve Dota 2 API such as the chat log and the status of the game every minute. The reason we chose to perform our experiments on an older dataset is because this dataset had a lot of additional fields that are not available straight from Valve's Dota 2 API. All the game from the Kaggle dataset come from the patch 6.85 of the game because patches affect the dynamics between heroes. Having data from different patches would cause inconsistency in the results. The game mode of these ranked matchmaking games is either ranked all pick, single draft, all random, random draft, or captains draft. Even if all the heroes are selected randomly in the all random game mode, players have the option to reroll their hero if they feel like the hero is unfit for the team or at a disadvantage against the other team. These game modes all provide the potential that any one of the heroes can be chosen. At late 2015, there were 112 heroes available to choose from. In this paper, we mainly focus on using the dataset from patch 6.85.

However, we also wanted to test the parts of our model that we could a more recent set of data. So we also gathered 67932 ranked matchmaking games from Nov. 24 to Dec. 1, 2017, directly from Valve's Dota 2 API using the python wrapper dota2api. We wanted to see how our methods stood up against more recent data. We gathered about ranked matches from patch 7.07c. Every match field contains the same information as the one from Kaggle minus the timestamped chat log and status for every player at every minute throughout the game. See **Table 6**.

Salient Features of the dataset

- **Match ID** - Unique Identifier about each game
- **Players** - List of 10 players where the first 5 player entries correspond to the ones on the Radiant and entries 6-10 correspond to the players on the Dire.
- **Player Entry** - Each player entry has information for that player in the match including what hero that player was playing, how much gold was spent by that player, xp per minute, gold per minute, kills, deaths, assists, the items purchased by the player at timestamps etc. The ones that we used in this study are hero_id, gold_spent
- **Game Outcomes and Characteristics** - Which side won, match duration, the status of various objectives, the status of gold and xp, at every minute throughout the game.
- **All chat log throughout the game** - Chat within each team is unavailable through replays and would be an invasion of privacy, so the dataset contained only the chat amongst all players. Each chat log entry contained the timestamp and player who sent the message.

Exploratory Analysis

Taking a look at the possible number of hero combinations we have $C(112, 10)$ possibilities. We read through the first couple of chat logs and saw that the losing team tended to be more toxic. So we decided to use chat logs as a feature. **Table 1** shows some analysis we did on different features of the dataset. The average game length of around 41 minutes and average gold difference between teams at the end of a game gave us the idea to see how impactful an early game advantage is for later stages of the game. Hence our decision to limit our study to the first 5 minutes and to consider metric such as last hits, XP, and gold.

Predictive Task

Our prediction task is to determine which team will win the game. We look at predicting which team will win the game before the game starts and looking into at how factoring in the first 10 minutes of the game will affect the prediction. We are trying to limit ourselves to predict this only by looking at the first 5 minutes of the game because over time, it becomes trivial to predict who will win because of potentially staggering differences in XP farmed, golds, key objectives, or item builds.

Average Game Length	2476.4535s
Absolute Average Gold Difference between teams at the end	22101.0035

Table 1: **Validation Accuracies: Chat based**

This is generally a hard task because there are 112 in patch 6.85 heroes and each hero is very unique and it's difficult to say which features define a good hero for the game. Some heroes excel at different things. The five heroes distribute amongst the 5 different positions in the game and the heroes are distributed in groups or individually across these lanes. There are also 50+ items in the game which cater to different builds of the hero over time. Some items are very situational and may be wielded as a counter to the other team's build as the game progresses. This makes the game more dynamic and the prediction task much harder.

While the kaggle dataset contains information about the entirety of each Dota match, predicting wins and losses based on some of the end-game data would be trivial. For example, the team with the most resources at the end of the match (gold, exp, etc.) were most likely the winners of the match. Thus, we wanted to limit our features to only use data from before the match starts, and data from early in the game. This way, we can make less obvious and more useful suggestions to players for increasing their win chances. These recommendations may include suggesting good heroes to pick, which early game actions to pursue, and possibly even how to behave during a match. Also, we did not consider kills, deaths, or assists (KDA) for each player because this metric does not a represent a player's skill and different hero roles can attribute to varying KDA.

Baseline

At the beginning the only thing we know is the hero picks. Having relevant match information about hero picks from previously labelled matches helps us establish a basic win rate that is solely based on team composition. This proved to be a basic predictor [1] but necessitated adding other features that are based on counters and team synergies. Our first objective was to predict wins and losses solely based on hero selection. First, we constructed a baseline where our feature vector was simply a one hot encoding of hero choice. For example, since there are 112 different heroes to choose from, our feature vector was of length 224 where the first 112 feature contained five 'ones' to represent the first team's hero choices, while features 113 to 224 represented the second team's hero choices.

We then ran this feature matrix through multiple classifiers and the best one - logistic regression yielded a validation accuracy of 60%. Logistic regression made the most sense to use and gave us the best result because the dependent variable in this case is just binary and most of our features just consisted of "nominal, ordinal, interval, or ratio-level independent variables" [5]. The training-validation split used was 40,000 to 10,000, and the validation accuracy was computed multiple times while the data was re-shuffled and re-split before each classification. We used the average of these validation scores to evaluate our model.

The feature representation for our baseline is $X[0:111]$ for Radiant team composition and then $X[112:223]$ for Dire.

$$\mathrm{X}_i = \begin{cases} 1 & \text{if Hero}_i \text{ is present} \\ 0 & \text{if Hero}_i \text{ is not present} \end{cases}$$

Since the baseline only encodes hero presence, this model will simply predict that the team with the highest summed individual-hero-win rate will win. The baseline does not consider how well heroes work together on a team or how well heroes on a team counter heroes on the other team. Therefore, we wanted to build a model that accounts for synergistic team compositions and counter-matchups.

Synergy Feature

To account for team synergy, we tried to construct a feature that describes how well each team fits the 'winning archetype combination' after mapping heroes to certain archetypes. In Dota 2, some heros scale better with more resources allocated to them and it is to the team's best interest to distribute gold and XP in such a way to maximizes the team composition's potential. For example, it is expected that winning teams generally contain these roles:

- **Carry (Position 1)** - The highest priority for levels (XP) and gold. Heroes that fall in this category are usually level or item dependant to scale later in the game to deal damage to enemy buildings.
- **Mid (Position 2)** - The second highest priority or XP and gold. The difference between position 1 and 2 is that position 2 generally builds toward items that give utility allow the team to engage in a favorable fight against the opposing team.
- **Offlane (Position 3)** - The third most dependent on resources usually build towards more expensive support items that support players will not be able to obtain in a reasonable amount of time.
- **Support (Position 4 or 5)** - The most resource starved role are not very dependant on XP or gold to make an impact in the match. Their resources are spent on pure support items that grant positional advantages on the play field.

By position, we mean the gold and XP allocation priority for each hero. Position 1 wanting the most resource distribution, and position 5 wanting the least. Our first step was therefore to map each hero to the role it is most likely to be played as. Conveniently, these roles tend to correspond with gold acquisition throughout the game. For example, heroes being played as a carry (position 1) tends to acquire and spend the most gold, while heroes playing the role as support (position 4 or 5) tend to make the least. Also, heroes that make moderate amounts of gold actually tend to be more flexible and can be played as either position 2 or position 3 roles. To dynamically assign heroes their most common role, we sorted the heroes on each team based on gold acquisition (0 being more supportive roles, 4 being carry oriented roles roles), and then we averaged the positions throughout the dataset to find each hero's most commonly played role. This preprocessing gave us a mapping of heroes to continuous values from 0 to 4 where a 3.7, for example, would indicate that the hero usually acquires the most the gold in each game and therefore tends to played as a carry role. It turned out that when combining this feature with the baseline, it had minimal affect on improving the results. This could be because in ranked matchmaking, players already attempt to distribute resources amongst team members appropriately and the baseline takes that into account.

Our feature representation for synergy:
$heroPositionRadiant = [Positionofhero[0:4]]$
$heroPositionDire = [Positionhero[0:4]]$
$X[224:229] = sorted(heroPositionRadiant)$
$X[230:234] = sorted(heroPositionDire)$

Hero Advantages and Counters

In addition to team synergy, we also wanted to account for counter-matchups and quantify how effective heroes on one team are against heroes on the other. While doing research on Dota 2 resources, we encountered a metric from Dotabuff.com called advantage. From each hero page, we used BeautifulSoup 4 to webscrape the advantage values for each hero against all other heroes. Advantage can be a negative value if the hero has a disadvantage against the other one. Advantage is also not symmetrical which means that one hero's advantage over another is not the same the other way around.
Advantage measures the matchup between two heroes regardless of their normal win rate. It is calculated by establishing their win rates both in and outside of the matchup comparing the difference against a base winrate. The calculation is procedural and advantage/disadvantage results are not designed to be symmetrical [6] .

The feature representation for representing hero

Baseline	0.6028
Synergy features	0.5408
Synergy + baseline	0.602
Advantage features	0.5192
Advantage + baseline	0.613
Synergy + Advantage + Baseline	0.6137

Table 2: **Validation Accuracies: Synergy + Counter matchups**

Baseline	0.6028
Number of chat entries per team	0.5294
Number of words per team	0.5381
Both	0.6197
Both with Baseline	0.6485
Toxicity	0.5614
Toxicity with Baseline and Talkativeness	0.6595 (0.6703 with MLP)

Table 3: **Validation Accuracies: Chat based**

advantages is:

$X[235 : 239] = \sum_{x\in R}\sum_{y\in D} advantage(x, y)$
$X[240 : 244] = \sum_{x\in D}\sum_{y\in Y} advantage(x, y)$

Table 2 represents the accuracies we achieved for the above mentioned features

Chat Logs

One unique thing about this database was that it contained the chat history for each match. This motivated us to examine whether or not there was any correlation between chat log history and winning a match. Since the chat log entries are analyzed after the match has ended, the goal behind using the entries as features is not necessarily to determine the features that yield the best win/loss prediction since chat entries might contain obvious indicators about who won. For example, someone from one team might congratulate the other team for winning, and while we could predict who wins based on giveaways like this, we would not really learn anything from this type of predictor. Therefore, the goal is to use the chat log to determine player attributes and behaviors throughout the match, and to see how these attributes affect win/loss outcomes. It should also be noted that this dataset only contains the "all-chat" logs which all 10 players can see, and not the "team-chat" logs which only players from the same team can see. The feature we constructed using the chat log is as follows:

$X[245]$= Word Count for Radiant
$X[246]$= Word Count for Dire
$X[247]$= Lines from Radiant
$X[248]$= Lines from Dire
$X[249]$= Toxic Word Count of Radiant
$X[250]$= Toxic Word Count of Dire

One metric we used as a feature can be interpreted as "team talkativeness." In our feature vector, we added the number of chat entries submitted per team as well as the total words typed per team. Using these features alone, we obtained a validation accuracy of 0.6197, and if we add the feature to the baseline, we end up with 0.6485 accuracy. Our interpretation of this boost in accuracy is that teams who are more talkative in all-chat probably tend to be more talkative in team-chat as well. Being talkative in team-chat is likely a good thing as the team would probably be more communicative in useful ways that help win the match such as by sharing enemy positions or by asking for help. Thus, we believe these results stress the importance of team communication during a match.

First Blood	0.561
First Tower	0.5961
First Aegis	0.7483
First Barrack	0.9169
All 4	0.9172
First 2 + baseline	0.6399
Toxicity with Baseline and Talkativeness	0.6595 (0.6703 with MLP)

Table 4: **Validation Accuracies: First team that captured first objective**

Baseline	0.6028
Baseline + synergy/advantage	0.6137
Baseline + synergy/advantage + Chat based features	0.6747
Baseline + synergy/advantage + Chat based features + Early objectives	0.6931
Baseline + synergy/advantage + Chat based features + Early objectives + Resources in first 5 minutes	0.7098

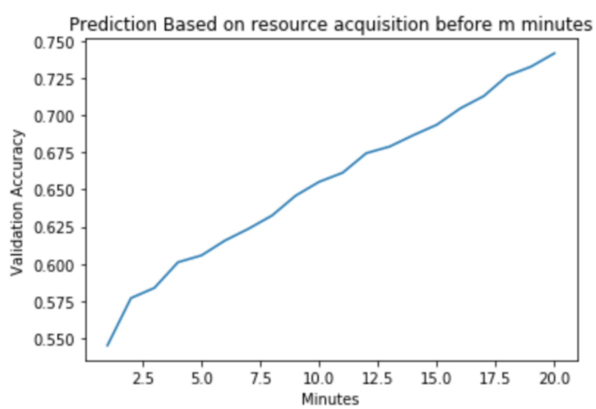
Table 5: **Summary of using different features**

Table 3 represents the accuracies using text features.

One flaw with this model is that it assumes that most messages in team-chat are valuable and constructive, but this is not the case. If a team member only says negative things about his or her teammates, this might hurt the overall performance of the team. This is known as being 'toxic'. Since our current talkativeness features did not distinguish useful messages from toxic ones, we wanted to add features to allow the classifiers to factor in toxicity. Our main approach was to have one feature per team that is the count of "bad words" spoken by each team using a previously defined list of bad words. The bad words list was taken from the League of Legends list of filtered words combined with manually adding less harsh, but still toxic, words that were not in this list. Although this method was chosen for implementation simplicity, this is not the best measure of toxicity as profanity is often used in positive or comedic ways, and negative remarks might be more subtle and not contain any obscene language. Still, we believe this method should suffice for detecting correlation between toxicity and team performance.

Early Game Markers

Finally, we wanted to explore the importance of early game activity by predicting win rate based on which team claims the first of each type of game objective, and also by predicting based on resource acquisition (gold and experience) within the first n-minutes. First, we made a feature based on which team obtained "first blood" which grants the player a large amount of gold for getting the first kill in the game. We then added features for other important objectives throughout the game including the first tower kill, the first team to get 'Aegis' (another important game objective), and the first barrack kill. Since the last two objectives are toward the end of the game, they are more obvious indicators of which team will win, and we therefore only use the 'first blood' and 'first tower' objectives in our model.



Second, we had features for the number of resources (one feature for each of gold, minion

Baseline	0.5784
Baseline + synergy/advantage	0.5973

Table 6: **Patch 7.07c Validation Accuracy: Synergy + Counter matchups**

kills, and experience per team) acquired before the 5 minute mark. We also observed the accuracy of our model over varying cutoff marks, but since we want to limit our predictions to use early game features, we stick to the **5 minute** mark for our final model features:

$X[251] = \sum_{x \in Radiant}$ Gold Spent(x)
$X[252] = \sum_{x \in Radiant}$ Last Hits(x)
$X[253] = \sum_{x \in Radiant}$ XP earned(x)
$X[254] = \sum_{x \in Dire}$ Gold Spent(x)
$X[255] = \sum_{x \in Dire}$ Last Hits(x)
$X[256] = \sum_{x \in Dire}$ XP earned(x)
$X[257] = \{1, 0, -1\}$ where First Blood is achieved by Radiant, None, Dire
$X[258] = \{1, 0, -1\}$ where First Tower is destroyed by Radiant, None, Dire

Table 4 shows accuracies achieved using the early game metrics.

Relevant Literature

As the game grew more popular and enthusiastic players were looking for more statistics to make use of, sites like dotabuff and OpenDota were able to provide some basic indicators and benchmarks for individualistic aspects of the play. Most of the data was directly pulled from Valve's API. However, the fast paced and dynamic nature of the game required deeper analysis of actions of players at different stages of the game. The original dataset was composed by Joe Ramir [7] to allow people to use data from individual plays and see how they affect the game.

Kinkade et al. [1] study some basic parameters from the above dataset and see how the affect the game. This was useful to us to understand what predictors have already been proved trivial and we were able to remove them from consideration or optimization. They measured synergy by win percentage with hero pairs on the team and hero countering based the win percentage of one hero compared to others on the opposing team.

Kevin et al.[2] use a queried format to predict match outcomes based on similarities to other games by querying and using methods like KNN for predicting outcomes based on how similar they are to previous weighted dimensions for similar matches.

Agarwala et al.[3] provide some information about using principal component analysis to team composition and showed that team composition although a good indicator is not a strong one but studying how metrics of team synergy and counters work, we were able to use our dataset and tweak the advantage feature to improve our overall accuracy.

Wang et al.[4] provide information about using a naive bayes classifier to predict outcomes solely based on team composition and probability that a particular lineup for a team resembles a "good" team composition.

Results and Conclusion

Our original feature representation just consisted of hero pool selection. We improved on this by adding team synergy features based on dynamic positioning and farming(Gold farming is directly correlated to the ability to buy items in the game [1]) feature. Our model is able to approach the classification task by trying to model the cohesion between the players themselves and the synergies between heroes picked on both teams and against each other. The chat log features also help model team cohesion, as we interpreted that players more talkative in all-chat communicated within their own team. The usage of the above diverse features was able to give us an improvement of 11% over the original baseline. The best accuracy on old patches achieved was 71%. **Table 5** shows the accuracy achieved using the different combination of features used. We constructed features from using chat log, unlike other methods from other literature and it turns out that there is some correlation between team communication and winning. Our results are significant because we demonstrate the importance of early game metrics and verbal behavior in addition to picking the right heroes.The synergy feature did not improve the baseline. It might be being covered by baseline or showed to be not as important as weighing the hero strength. A similar occurrence happened with using PCA to determine good hero compositions [3]. All of the other features contribute to improving the accuracy because they are not as related to each other as team composition and hero selection.

Future Work

While training we were able to use the chat log to enhance our feature vector. It would be interesting to mine data for team communication maybe from some voice server like discord or from some VoIP server. This would provide richer context of processing the data. It is also possible to extract match cues that could potentially enhance our prediction. Currently no mathematical model exists to map item builds or positional presence of heroes to game impact. Further work could improve our features of position based team synergy and item builds to improve better accuracy. The feature representation mentioned in the results above could be used in conjunction with existing bots (for example OpenAI bot) to provide tips to playing users during scrims or training sessions.

Citations and References

[1] **DOTA 2 Win Prediction** - Kinkade, Nicholas and Kevin Lim, Kyung Yul.

[2] **How Does He Saw Me? A Recommendation Engine for Picking Heroes in Dota 2** - Conley,Kevin and Perry, Daniel.

[3] **Learning Dota 2 Team Compositions** - Agarwala, Atish and Pearce, Michael.

[4] **Outcome Prediction of DOTA2 Based on Naive Bayes Classifier** - Wang, Kaixiang and Shang, Wenqian.

[5] **statisticssolutions.com**

[6] **dotabuff.com**

[7] **https://www.kaggle.com/jraramirez/dota-2-matches-dataset**