LoLIntelligence: Alpha Development

Alyssa Adams*

Arizona State University, Tempe

amadam15@asu.edu

Abstract

This paper reports on the progress of the project LoLIntelligence as it undergoes its first round of development.

I. Introduction

League of Legends is an online team-based strategy game that has become immensely popular over the last 5 years (cite) and is currently the most popular game of its type (cite). As of 2015, there are 4 million active players (players who log in at least once per day) across the globe and this population has been steadily increasing since. As such, the game's developers have made all the data generated from gameplay available to the public, specifically for application developers. These applications are used by players to analyze their own games and statistics, as well as other team members to set correct expectations for potential gameplay. For example, the website LolKing.com will return the average winrate of a player, as well as the winrates for specific champions (Figure 1). Often times, players will use web applications like this to learn about their team mates before a game begins.

There are several more examples of applications that are designed to access game data in an easy-to-use way. However, none of these applications analyze time series data for a large ensemble of players, nor in ways that involve information-theoretic methods. These measures are useful in determining which factors change gameplay over time, as well as how player habits affect the overall play experience. Since the data dates from 2014 to current, it is surprising that this extremely rich and detailed dataset has gone untouched for network and information-theoretic analysis.

This paper reports on a first attempt to analyze the data generated by League of Legends in an information-theoretic sense. Because the data is so vast, only a very small subset of data is first considered to develop methods of measurement and analysis. This data is translated into a time series of Boolean values that is assumed to be generated by a dynamic Boolean network that changes nodes and edges over time. The time series is used to calculate information storage and information processing within player behavior. Finally, the outcomes of player behavior is used to determine how the Boolean network changes over time.

Many of the assumptions made throughout this paper are not based on any real fact and are solely from many (too many) hours of gameplay experience.

II. Gameplay

To gain a sense of which parts of the data are important for which measure, it is worth outlining how players interact with League of Legends. If a user wishes to play a game, they may queue up with friends, else they are placed with other random players. Two teams of five players compete against eachother to win the game.

Before the game starts, both teams have a change to "ban" and pick particular champions. Champions are characters that players will play, and each player may only choose

*Expert Teemo player (Gold V)

one champion to play throughout the course of a single game (Figure 2). If a champion is “banned”, nobody on either team will be able to play that champion, thus eliminating it from the game. Only one player can play a given champion such that duplicate champions are not allowed in a single game.

In addition to picking a champion, players must also pick a position on the team. Positions also indicate where on the game map (see Figure 3) that player will spend most of their time (top of the map, middle of the map, etc, and are named accordingly: top, middle, etc.). The concept of positions is not unlike basketball; each of the 5 members on a team has a particular role that they fullfill. Players find that some champions are good with many positions, while some are only good with a few.

Teams pick their champions in a set order, and the first team is randomly decided by the game:

- Red team randomly gets to pick their first champion
- Blue team picks two champions
- Red team picks two champions
- Blue team picks two champions
- Red team picks two champions
- Blue team picks their last champion

After teams have finished picking their champions and positions, the game begins on a 2D area map. One team is stationed at one corner of the map, and the other team is stationed on the furthest, most opposite corner (Figure 4). The goal for each team is to destroy the other team’s base by earning gold throughout the map and buying items to make their champion stronger than the enemy champions. Games can last anywhere from 20-60 minutes and end when one team surrenders or loses their base.

Ranked games place players on a skill tier system based on how many games they have won. Bronze tier is the lowest skill players, while Challenger tier is for ESports players. Winning many games over time increases the chances a player will climb the tier, and get shiny emblems on their profile. Players are always randomly matched with players in the same tier.

Figure 1: *Screenshot from LolKing.com. Here, the statistics for the player "gigglepuss" are displayed. Any player’s username can be used to return data for that specific player.*

Figure 2: *One of League of Legends' most beloved champions– Teemo. Teemo can be played in 4 out of 5 positions on a team well (according to current overall player consensus).*

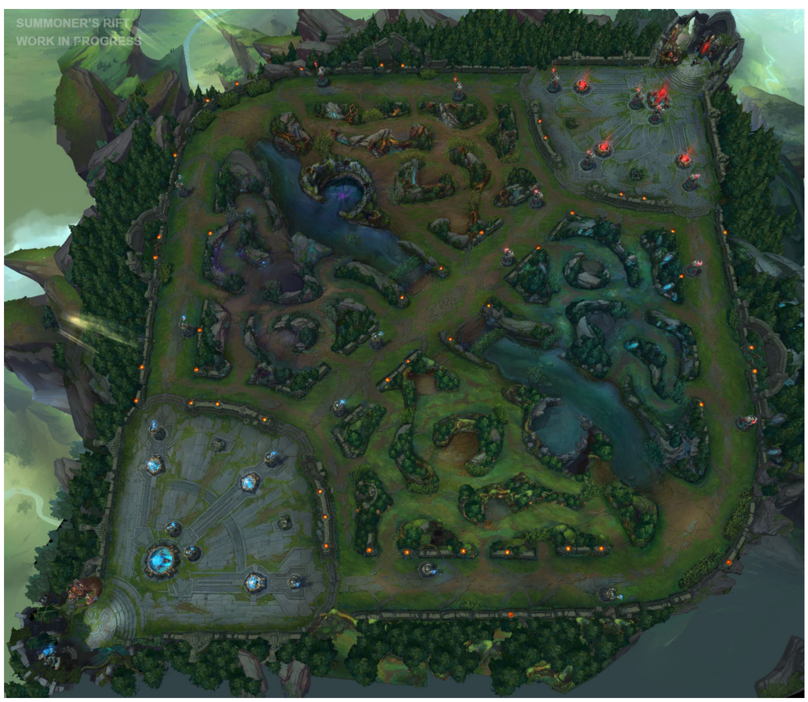

Figure 3: *The main 2D map where players interact throughout the game (cite). Team bases are highlighted in grey.*

III. Actual Gameplay

League of Legends is notorious for having a toxic player community. Efforts are constantly enforced to minimize the effects of toxic players by the game developers, which has had some success. From a behavioral standpoint, it is easy to notice the structure of the game is actually deeply affected by player behavior. For example, toxic players cause many players to surrender games early, which affects the overall statistics of the game. Toxic players often lose much more than other players, thus placing them in lower tiers. This makes it extremely difficult to climb out of lower tiers, and any trends seen in lower tiers are generally considered "bad gameplay strategies". Therefore, the majority player community avoids particular gameplay techniques because of their lower-tier association, changing the gameplay across all skill tiers.

By far, toxicity is not the only behavior that affects gameplay. Professional players set many trends that most other players work to imitate. Using Teemo as an example, one professional player started playing Teemo more like a mage, while other players were still playing him like a tank. Since many people watch this player's behavior, many imitated his Teemo style and the trend currently continues.

When a particular champion becomes popular, players general learn the strategy to beat it with a counter-strategy. Once the counter-strategy is popular, then the counter to that counter-strategy becomes popular, and so on. The game is not static by any means, and changes according to feedback to player behavior.

IV. Creating a Network

All the above can be represented as a network in several different ways, each highlighting an aspect of the game. Intrinsically, there is no concept of one champion being better than another champion at any given time. However, the popularity of a given champion and current trends creates a dynamic heirarchy of preferred champions. If Teemo becomes very popular, many players will assume the champion is better than the rest. This may not reflect on the actual winrate of players who play Teemo, however. At any given time, a network can be instructed to show which champion beats which champion based on the outcomes of players who play those champions. According to counterstats.net, Teemo beats Renekton in the middle position because Teemo has the higher winrate for that position. These statistics can be found for every champion in every position. This network is shown in Figure 5, where champions are nodes and directed edges indicate which champion beats what.

An immediate question to ask is whether or not the vertex out-degree of a champion correlates with that champion's overall winrate. Figures 6 and 7 compare these two measures in color, where red indicates high values and blue indicates low values. Notice how the two measures seem to be much less correlated than intuition would lend.

Recall that these networks are not static. Every few months, a new node is introduced to the network since the developers release a new champion. In addition, old champions are changed or updated, items are added or changed or removed, or spatial properties of the map are altered. All these changes affect this network drastically. To avoid extreme network changes, only data between two specific game patches will be used.

To further reduce the data set, only the last 50 games played by myself will be used. This automatically introduces a strong bias towards the playstyle of a single player, but the idea is to develop scalable measures to eventually analyze much larger ensembles of games.

V. Information Measures

Active information and transfer entropy are Shannon-based ways of measuring information storage and processing (respecively) in a system. They also do not require more than a time series of data to calculate, which is a convenient measure to start with for this data.

Figure 4: *A network that shows how champions (nodes) beat (directed edges) other champions.*

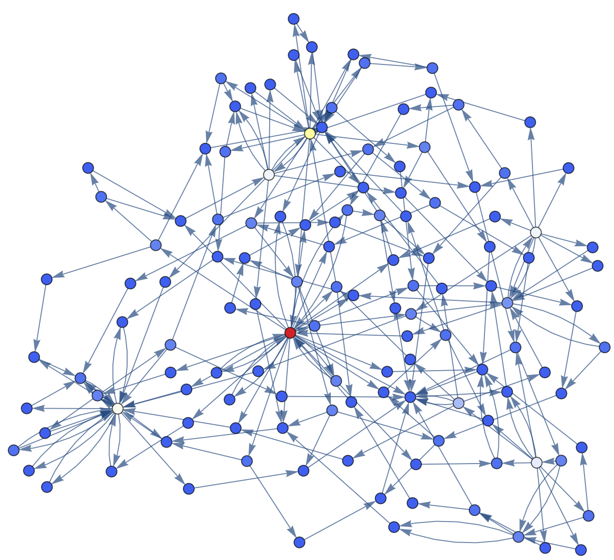

Figure 5: *The same network in Figure 5, except vertices are colored according to out-degree values. Blue color indicates low values, while red color indicated high values.*

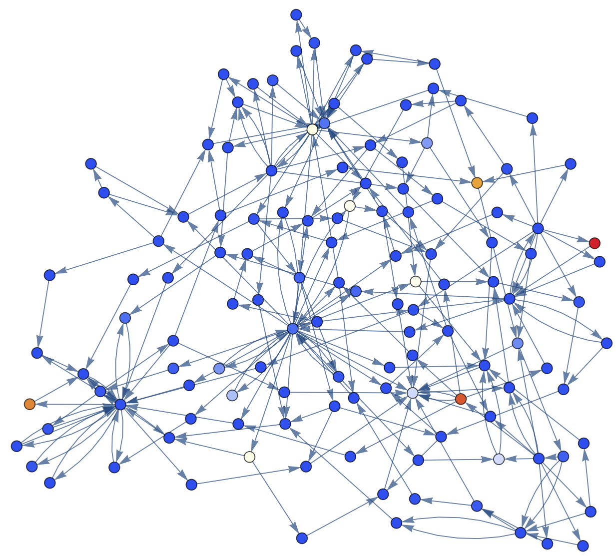

Figure 6: *The same network in Figure 5, except vertices are colored according to the corresponding champion's winrate at the time this network was generated. Blue color indicates low values, while red color indicated high values.*

Since the subset of data being considered is so small, no complete networks (one where all champions are present) can be extracted. As a basline to generate a time series, the data was sorted into days as a time unit. Two types of values for the time series were considered: champion winrate and champion popularity. The winrate was only considered between that champion and their opponent with the same position. In other words, even if Teemo's team lost the game, if Teemo beat his opponent in the middle position by earning more gold, then it is counted as a win for Teemo. If Teemo wins more games than he loses in a day, then the Teemo node is assigned a 'winrate = 1' value for that day, otherwise the node is assigned a 'winrate = 0'. Similarly, if Teemo was in more than a third of all games played that day, then that node is assigned a 'popularity = 1', otherwise it gets a 'popularity = 0'. These thresholds were arbitrarily chosen, and need further justification by investigation.

Given two types are states, there are two seperate time series' for this subset of data. Both the active information (AI) and transfer entropy (TE) were used on each time series. Both AI and TE depend on a history length k, which indicates the number of timesteps necessary to perform the calculation. The higher the k value, the more timesteps are used to analyze information dynamics. It is analogous to memory required to process or store information.

VI. Results

To gain a general sense of the information dynamics in this data, both the AI and TE were calculated for each time series (Figures 7-10). The values are ranked by frequency and different k values are indicated by color. In general, AI increases with larger k values, while TE decreases with larger k values. AI values are much larger than TE values in both time series'. Overall, this general information analysis reveals little difference between the two types of time series'.

If TE is calculated only for nodes that are connected by an edge, the distribution of TE values changes a bit more between the two time series' (Figures 11 and 12). On average,

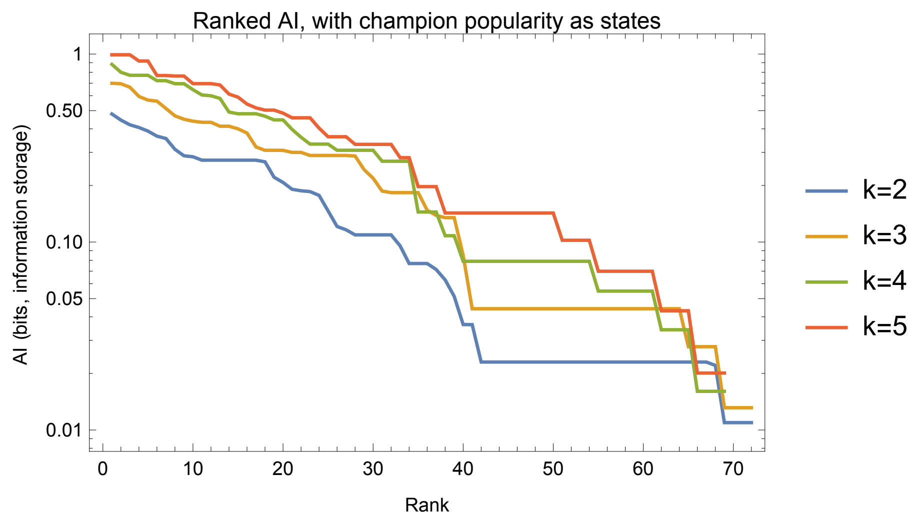

Figure 7: *The distribution of active information (AI) values are sorted by rank for the popularity time series. Different colors indicate different values of k.*

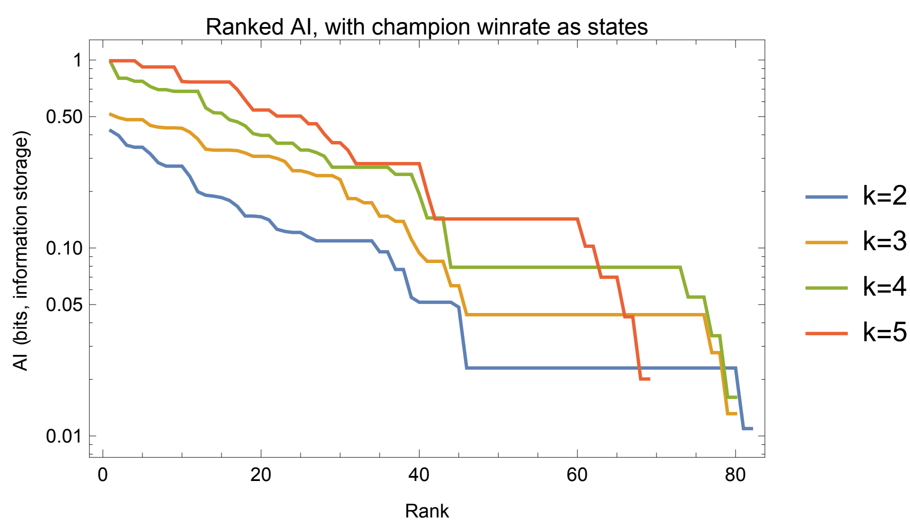

Figure 8: *The distribution of active information (AI) values are sorted by rank for the winrate time series. Different colors indicate different values of k.*

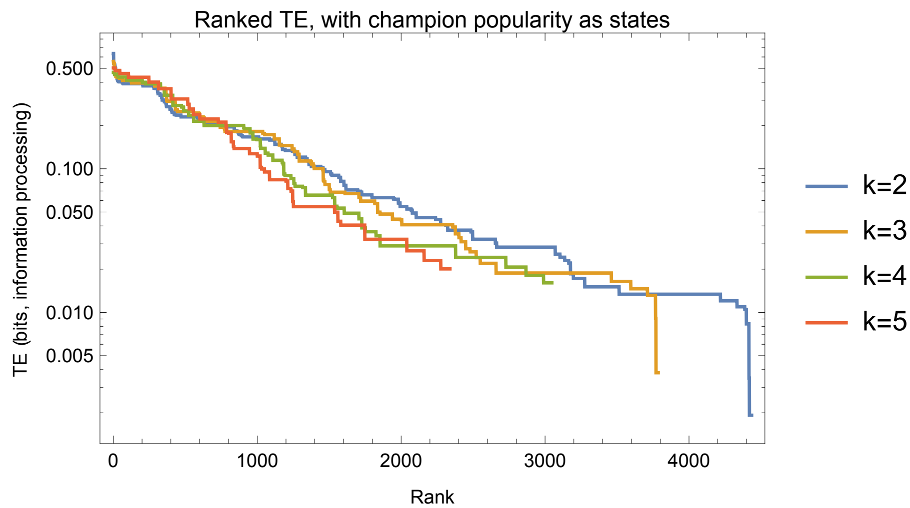

Figure 9: *The distribution of transfer entropy (TE) values are sorted by rank for the popularity time series. Different colors indicate different values of k.*

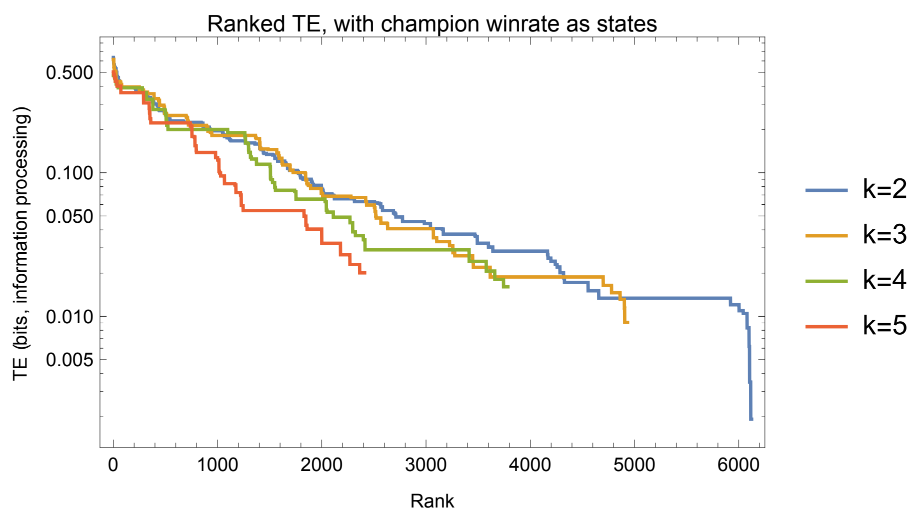

Figure 10: *The distribution of transfer entropy (TE) values are sorted by rank for the winrate time series. Different colors indicate different values of k.*

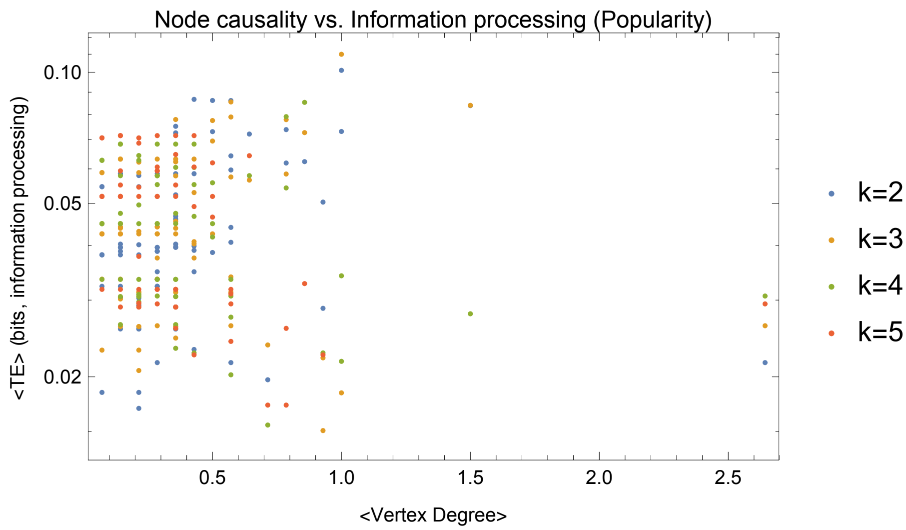

Figure 11: *Average node degree (since node conncetivity changes over time) vs. that node's average transfer entropy value. This is over the popularity time series. As before, different values for k are shown by color.*

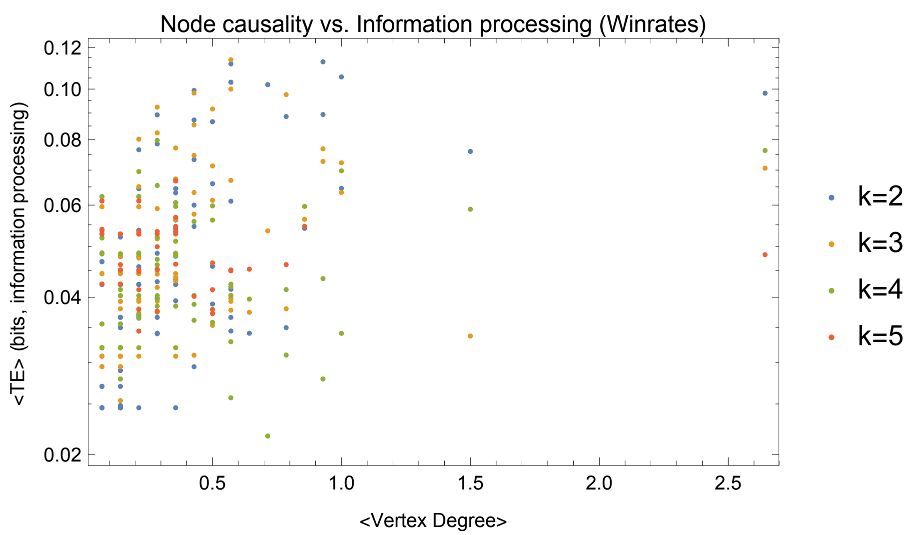

Figure 12: *Average node degree (since node conncetivity changes over time) vs. that node's average transfer entropy value. This is over the winrate time series. As before, different values for k are shown by color.*

TE values are slightly higher for winrate states. Most TE values are found between edges with a low average degree.

Because Figures 11 and 12 lack a clear trend, it is more informative to plot the total TE processed in each dataset as a function of k. Figures 13 and 14 show a clear trend, and as seen in Figures 9 and 10, the total amount of information processed over a time series decreases as k increases. The effect is more dramatic in the winrate time series.

Finally, it will be useful to see the trend between AI and TE. Figures 14 through 18 show a heatmap of TE values as a function of AI. Nodes are arranged on the x and y axes according to their AI value. Nodes with low AI are towards the origin, while nodes with high AI are away from the origin. Both the x and y axis have the same arrangement for nodes. The distribution of the nodes' AI values is shown below the heatmap and it only shown for the x axis since it is the same in the y axis.

TE values are not symmetric, so the heatmap is not symmetric. The TE for a node A using B is different than node B using A. Each Figure shows data for a different k value. These figures only represent data from the winrate time series, since the same figures produced by the popularity time series were not drastically different.

VII. Discussion

The most striking results are shown by Figures 13 and 14. They indicate that information processing occurs at a greater rate with a shorter memory. The effect is even greater in the winrate time series. This might indicate that players process the most information about winrates in the last few days. This is also true about popularity.

In general, all the figures show that players store a lot of information in both winrates and popularity, but do not process data as much. Nodes that have low information storage are generally nodes that process the most information, but this assumption gets more obscure as the memory increases. Figure 18 is a great example of this. Note how the lines between information processing and storage begin to blur.

Or course, these effects may be artifacts of a tiny sample size. The play habits of the single player are geared more towards playing Teemo over and over again, no matter the win or loss outcome. The easiest way to climb the tier ladder is to only play champions that have high winrates. This data is biased against that general trend.

VIII. What's next?

The algorithms delevoped for this alaysis are certainly scalable with data size, so the most immediate step is to use data from a representative sample of the League of Legends community. Several questions are worth exploring to understand how this network behaves. Here are a few questions, ordered by tractibility:

- How do these results change with different thresholds for winrate and popularity?
- How do these results change with a representative sample of the entire League community?
- How do these results change as a function of player skill (sampling from each tier separately)?
- How do these results change between patches per tier, specifically when a new node is added?
- Are there nodes that have consistent AI and/or TE values?
- What is the most sensitive part of the network, and does it change over time?

The goal is to understand how League of Legends behaves as an ecosystem of player behavior. It seems like players perform a distributed computation to change the network, and then respond to those changes in a state-dependent way. State dependence is the central dynamical driver of the network, and with the game developers constantly adding nodes and changing properties of the game, the network never falls into an attractor state. The game is specifically intended to avoid attractor states.

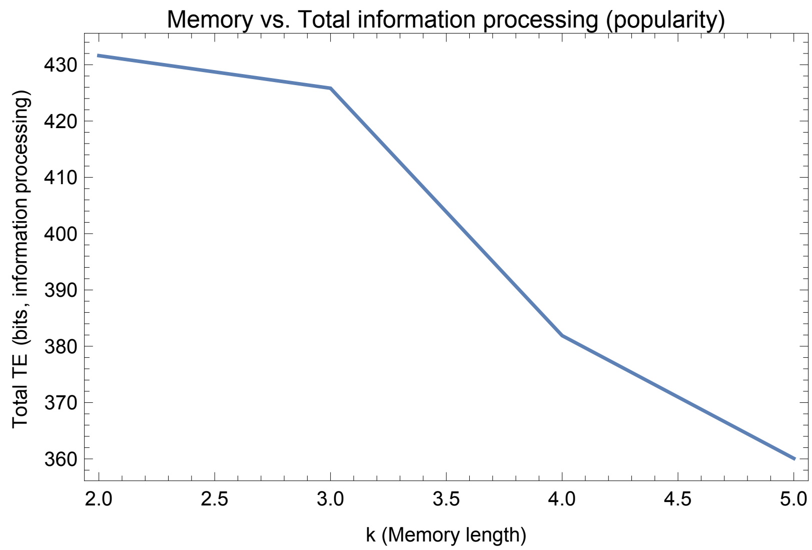

Figure 13: *One of League of Legends' most beloved champions– Teemo. Teemo can be played in 4 out of 5 positions on a team well (according to current overall player consensus).*

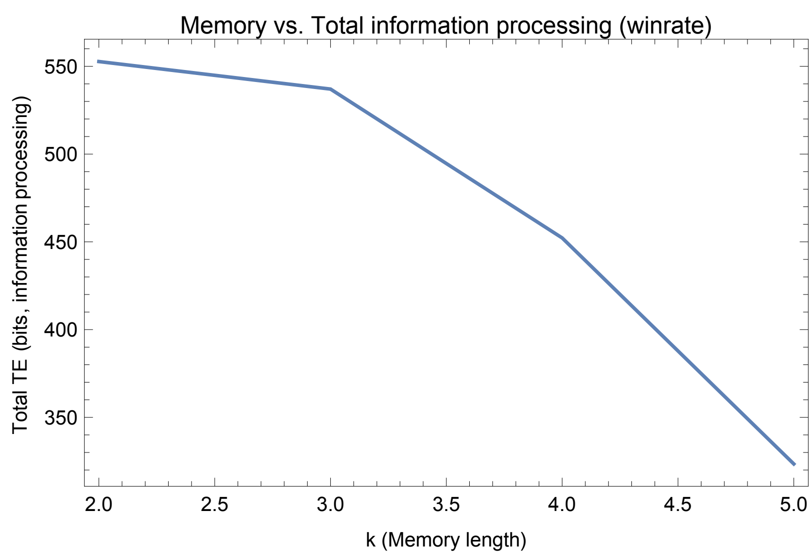

Figure 14: *One of League of Legends' most beloved champions– Teemo. Teemo can be played in 4 out of 5 positions on a team well (according to current overall player consensus).*

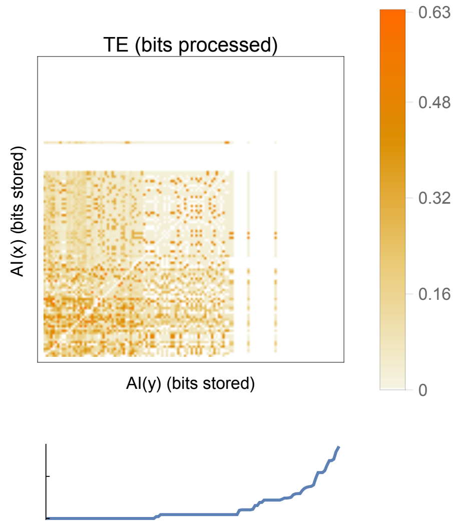

Figure 15: *Heatmap of TE values between nodes for k=2 from the winrate time series. Nodes are arranged on the x and y axes according to their AI value. Nodes with low AI are towards the origin, while nodes with high AI are away from the origin. Both the x and y axis have the same arrangement for nodes. The distribution of the nodes' AI values is shown below the heatmap and it only shown for the x axis since it is the same in the y axis.*

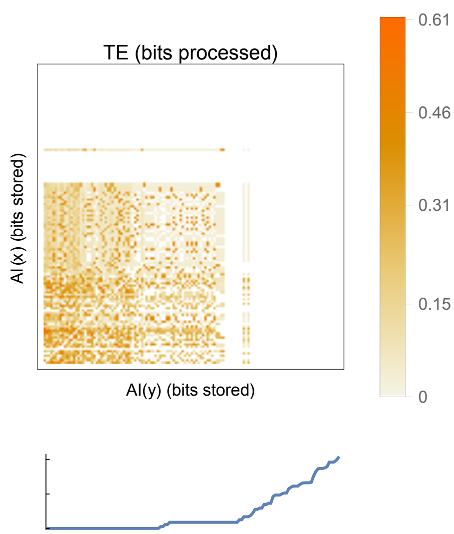

Figure 16: *Heatmap of TE values between nodes for k=3 from the winrate time series. Nodes are arranged on the x and y axes according to their AI value. Nodes with low AI are towards the origin, while nodes with high AI are away from the origin. Both the x and y axis have the same arrangement for nodes. The distribution of the nodes' AI values is shown below the heatmap and it only shown for the x axis since it is the same in the y axis.*

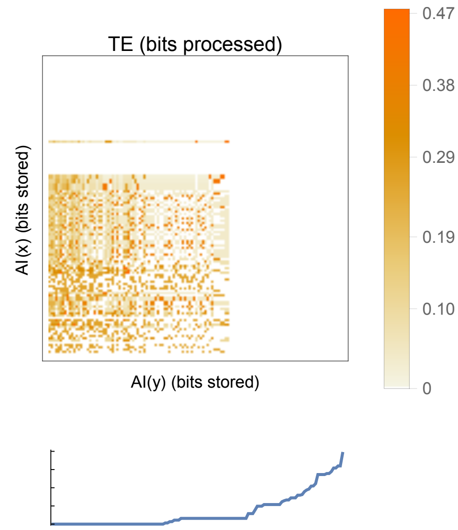

Figure 17: *Heatmap of TE values between nodes for k=4 from the winrate time series. Nodes are arranged on the x and y axes according to their AI value. Nodes with low AI are towards the origin, while nodes with high AI are away from the origin. Both the x and y axis have the same arrangement for nodes. The distribution of the nodes' AI values is shown below the heatmap and it only shown for the x axis since it is the same in the y axis.*

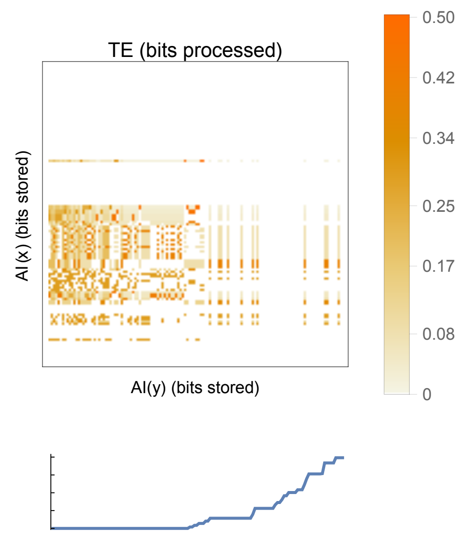

Figure 18: *Heatmap of TE values between nodes for k=5 from the winrate time series. Nodes are arranged on the x and y axes according to their AI value. Nodes with low AI are towards the origin, while nodes with high AI are away from the origin. Both the x and y axis have the same arrangement for nodes. The distribution of the nodes' AI values is shown below the heatmap and it only shown for the x axis since it is the same in the y axis.*

Understanding how the game evolves with topology and responds to changes can help predict future dynamics and player strategies. This type of analysis on a network can be extended to other social networks or security systems. Either way, this analysis will be continue to develop and be processed in a web application where users can learn about information-theoretic techniques, and how League of Legends can be used to study human behavior and distributed computation.

References

[Sample reference: Figueredo and Wolf, 2009] Figueredo, A. J. and Wolf, P. S. A. (2009). Assortative pairing and life history strategy - a cross-cultural study. *Human Nature*, 20:317–330.