TEAM TEEMO

Interactive Visualization of Team Compositions Using Association Rules in League of Legends

Jos Vilier, Cheukming Ku, Hanbit Kim, Johnnie Rex Cairns, Hyeong Sun Im, Christopher Dae-Hyun Kim

Summary

The online game League of Legends is played by over 100 million players worldwide but currently there is a lack of easy-to-use, team-oriented analytical tools to improve player performance & experience.

The game consists of a drafting and fighting part. Most players focus on the fighting part because that is the most exciting, but the drafting part has a very large impact on the game's outcome. We have created an interactive, intuitive, and flexible visual tool that helps players draft winning champion combinations to improve their game. Recommendations are created by extending association rules to graph theory.

Champion Roster

Data collection

Data was collected from the **Riot Games API**. Riot Games is the developer of League of Legends and operates all infrastructure to host matches.

Match data is **segmented by geographic region** (Japan, Korea, Europe West, Europe Nordic & East, and North America) and contains temporal features, such as time/date of game played and game patch number. We extracted more than **1 million match results** from the API with cleaned data being **~1 gigabyte** in size.

Hello, we would like to have match data for match #1337 in region EU west

Match #1337 in EU West: {Team 1: [MissFortune, Ezreal, Lucian, Nami, Braum], Team 2: [Yasuo, Ziggs, Gwen, Nunu, Samira], Winner: Team 1}

TEAM TEEMO

Thank you, what about match #420420 in Korea?

RIOT GAMES

Match #420420 in Korea: {Team 1: [Lillia, Sejuani, Morgana, Ashe, Varus], Team 2: [Velkoz, Bard, Xerath, Sona, Darius], Winner: Team 2}

Approach 1: Strong Champion Pairs

How does one draft a strong team? Our first interactive graph helps with selecting **strong combinations of two champions** and has directed and weighted edges**.** The main characteristics are **champion popularity**, the **win-rate of the champion pairs,** and the **confidence of champion pairs**.

Confidence is the conditional probability between champions.

$$conf(A \to B) = P(B \text{ is in team} | A \text{ is in team}) = \frac{P(A \cap B)}{P(A)}$$

Data for the graph was processed with a minimum **support**, **confidence, lift, and pairwise win-rates**. Lift is a measure of dependence between the champions where a value greater than 1 indicates positive dependence.

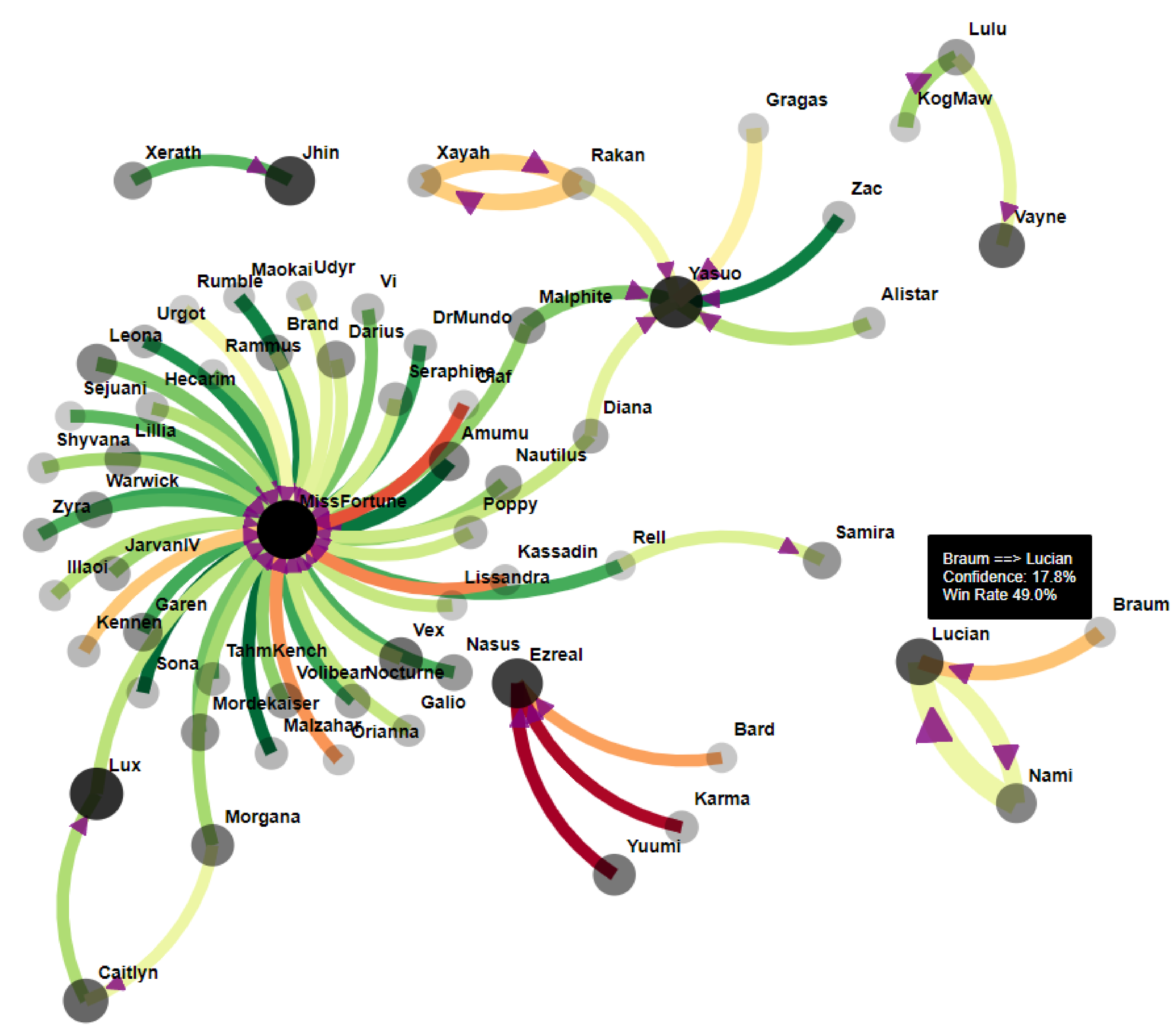
$$lift(A \to B) = \frac{P(A \cap B)}{P(A)P(B)}$$

Popularity is given by node size and opacity whereas confidence is given by edge thickness and tooltip. Win-rates are indicated by a diverging color scale where low, average, and high win-rates are respectively indicated by red, yellow, and green. Arrows indicate edge direction and the antecedent/consequent of a particular rule.

Interactivity allows the user to select and filter by region, specific champions, minimum confidence, and minimum win-rate.

The graph has shown that popular champions tend to have higher degrees. The graph shows that Miss Fortune is the most popular champion, but several combinations including the champion have a low win-rate, such as with Kassadin. Yasuo is less popular, but is associated with high win-rate pairs, such as with Zac.

Strong Champion Pairs Graph Example



Approach 2: Strong Team Compositions

Our second interactive graph helps with selecting **strong communities of champions determined by community detection**. Communities were determined using greedy modularity maximization from the Python package NetworkX. It is an undirected, weighted graph using **support as edges**.

Support can be referred to as the probability of champion pairs occurring.

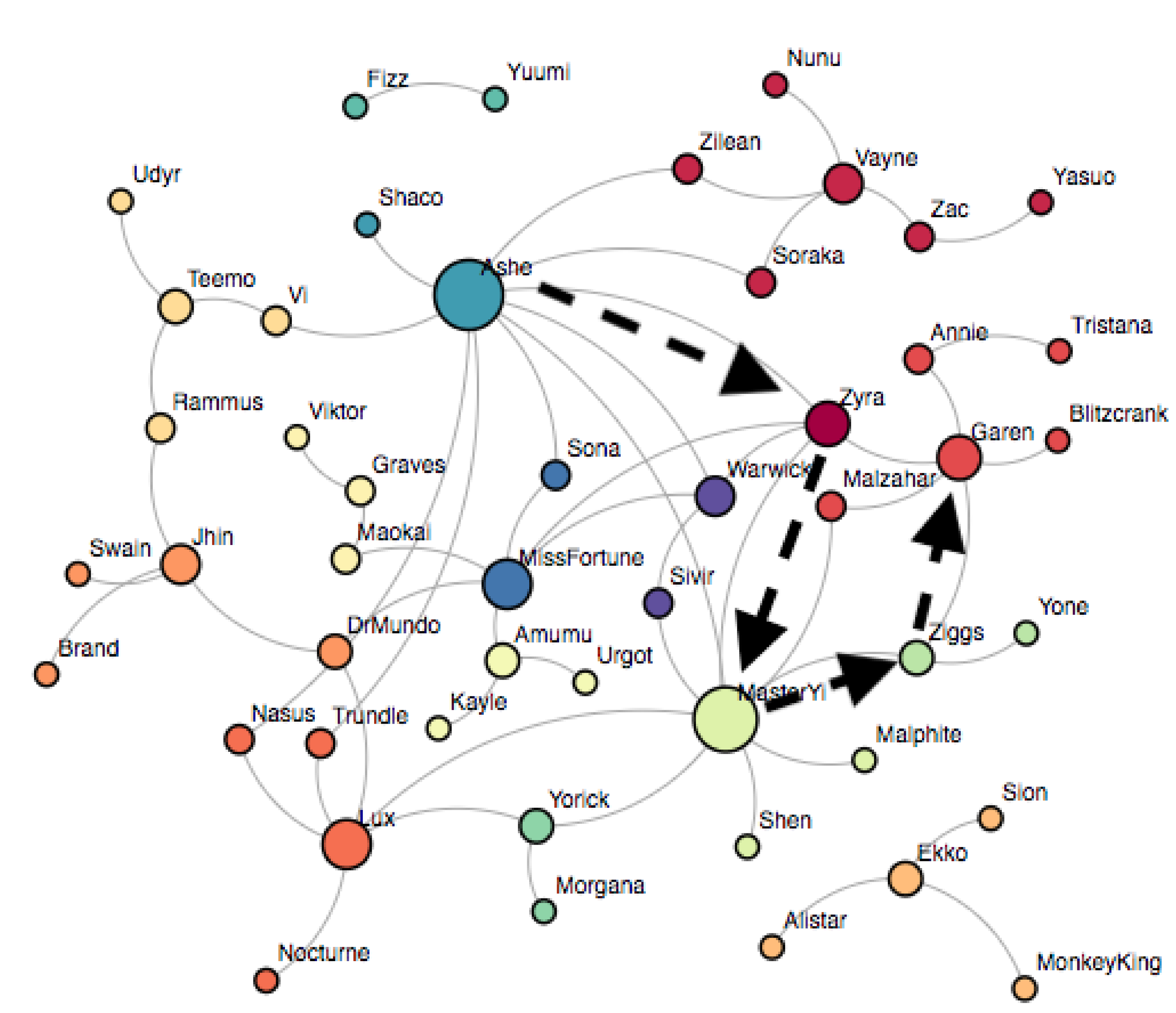
$$supp(A \to B) = \frac{|\text{A and B}|}{|\text{Teams}|} = P(A \cap B)$$

Data for the graph was filtered using a minimum support = {0.1%, 0.3%}, minimum confidence = 5%, lift > 1, and pairwise win-rates > 55%.

Node size is proportional to node degree, and node color is associated with membership to a community. Interactivity allows the user to select and filter by region. Users can drag the nodes to improve graph clarity.

Team recommendation is completely user derived while our analysis provides strong and sophisticated recommendations. To select a team, users can travel from node to node due to the bidirectional nature of the support with or without regard to community membership insofar there is a viable pathway.

Strong Team Composition Graph Example



Results

A survey was posted on the forum for League of Legends on the social media platform Reddit.

Out of **22** respondents, **95.5%** stated that team composition was an important factor when selecting a champion and **82.2%** have stated that they have not seen other analytics providers using sophisticated methods to determine strong team compositions. Graph ease of use received a mean rating of **3.7** with a median of **4.0** while graph effectiveness received a mean rating of **4.0** with a median of **4.0;** all ratings were from 1-5, with 5 being "high"**. 77.3%** have stated that they would both use the graphs as a planning tool and believe that the graphs would increase their chances of winning.

The model and results can be improved to segment data further by player operational skill level and game patch number. In fact, **100%** of respondents would like to see a patch specific filter, **86.4%** would like a player skill level filter, and **77.3%** would like to see a role specific filter.