ggpyparser: A Python Library for Esports Data

Lou Zhou
Rice University
lz80@rice.edu | lou-zhou.github.io

RICE UNIVERSITY Sport Analytics

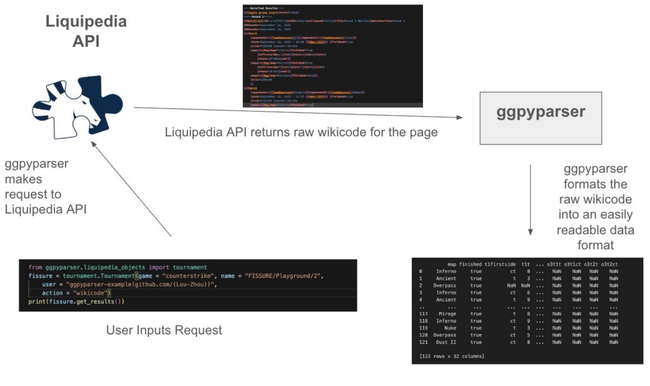
Code & Documentation:

Motivation

- With the sudden growth of electronic sports (esports) industry, there is now new demand for reliable and accessible esports data
- Despite the digital environment, besides the largest titles, the general public does not have easy access to regularly-updated and easy to use data without having to rely on private data companies or web scraping
- In addressing these challenges, ggpyparser was built to provide easy to access and regularly-updated data across many different esports

Methodology

- In gathering data, ggpyparser uses the API provided by Liquipedia, a community run, open-access esports encyclopedia covering a wide variety of esports, similar to Sports Reference
- By parsing Liquipedia's standardized markup language, wikicode, ggpyparser is able to generate detailed and regularly updated data from over 55+ esports, describing players, teams, competitions, and the matches within those competitions
- To increase ease of use, ggpyparser takes an object-oriented approach where the three main types of Liquipedia pages: tournaments, players, and teams, are represented as objects
  - From these objects, users can pull data from these pages using methods that return structured outputs like dictionaries, lists, or pandas DataFrames
  - To mitigate rate-limiting, ggpyparser supports parsing up to 50 different pages from one call



Titles Covered

Table 1: Titles Covered by `ggpyparser` (MOBA and Shooter Titles)

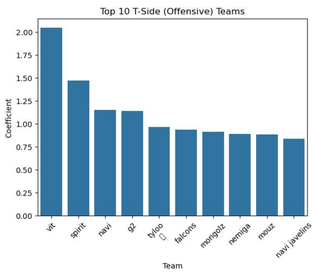
Multiplayer Online Battle Arena Games	First Person Shooters	Other Shooters
Dota 2	Counter-Strike	PUBG
Mobile Legends	Valorant	Fortnite
League of Legends	Apex Legends	Free Fire
Honor of Kings	Overwatch	Splatoon
Brawl Stars	Rainbow Six Siege	
Wild Rift	Call of Duty	
Heroes of the Storm	Crossfire	
Deadlock	Halo	
Smite	Team Fortress 2	
	Quake	
	Doom	
	Splitgate	
	Critical Ops	
	Tarkov Arena	

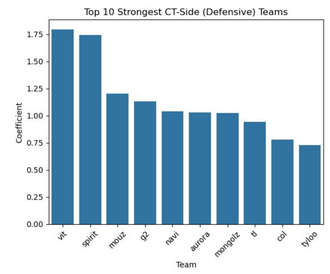
Table 2: Titles Covered by `ggpyparser` (Sports, Fighting, Strategy, Other)

Sports	Fighting	Strategy	Other
Rocket League	Street Fighter	Starcraft I/II	World of Tanks
EA Sports FC	Tekken	Age of Empires	Warcraft III
Sideswipe	Mortal Kombat	Clash Royale	Hearthstone
Rematch	Smash	Stormgate	Teamfight Tactics
Omega Strikers	Marvel Rivals		Pokemon TCG / VGC
GOALS	Brawlhalla		Trackmania
	Naraka: Bladepoint		GeoGuessr
			osu!
			World of Warcraft
			Tetris
			Wildcard
			Sim Racing
			War Thunder

Potential Usage

- To exemplify potential uses of the data from this package, we estimate a team's strength in the esport Counter-Strike 2 for both the offensive side ("T-Side") as well as the defensive side ("CT-Side")
  - To build these estimates, we build a random effects model using offensive and defensive teams as random effects to predict score difference
  - Because the layout of a level (a "map") can naturally favor one side, we also include the map itself as a factor in the analysis
- From this analysis, we can generate the best attacking and defending teams





1. For readability, we take the magnitude of each team's coefficient

Example Package Functionality

Getting a Team's Roster History

```
from ggpyparser.liquipedia_objects import team
sample = team.Team(game = "apexlegends", name = "Cloud9",
    user = "ggpyparser-example(github.com/Lou-Zhou)",
    action = "wikicode")
print(sample.get_players().head(5).to_string())
```

✓ 0.6s Python

	flag	id	name	joindate	leavedate	newteam	newteamrole	team	role	newrole	for	forflag	forname	tournament
0	us	PVPX	Jamison Moore	2019-02-22	2019-??-??	Cloud9	Coach	NaN	NaN	NaN	NaN	NaN	NaN	NaN
1	us	Chappie	Justin Andrews	2019-03-21	2019-??-??	Cloud9	Streamer	NaN	NaN	NaN	NaN	NaN	NaN	NaN
2	us	Grego	Gregory McAllen	2019-03-01	2019-08-07	team secret		NaN	NaN	NaN	NaN	NaN	NaN	NaN
3	us	Frexs	Joseph Sanchez	2019-03-21	2019-08-07	nrg esports		NaN	NaN	NaN	NaN	NaN	NaN	NaN
4	us	Overpowered	Timothy Liang	2019-04-17	2019-08-07	retired		NaN	NaN	NaN	NaN	NaN	NaN	NaN

Getting All Match Data for a Competition

```
from ggpyparser.liquipedia_objects import tournament

    #initializing the object and making the request to the Liquipedia API
sample = tournament.Tournament(game = "brawlstars", name = "Brawl_Stars_World_Finals/2024",
                               user = "ggpyparser-example(github.com/lou-zhou)",
                               action = "wikicode")
print(sample.get_results().head(5).to_string())
```

✓ 0.8s Python

	map	maptype	firstpick	score1	score2	t1c1	t1c2	t1c3	t2c1	t2c2	t2c3	t1b1	t1b2	t1b3	t2b1	t2b2
0	Belle's Rock	Knockout		1	2	1	R-t	Moe	Darryl	Sprout	Belle	Rico	Gene	Gus	Kenji	Tick	Angelo
1	Center Stage	Brawl Ball		1	0	2	Gale	Buster	Stu	Rico	Otis	Jacky	Nita	Kenji	Max	Surge	Clancy
2	Shooting Star	Bounty		2	0	2	Gus	Gene	Kenji	Piper	Tick	Pearl	Byron	Angelo	Belle	Max	R-t
3	Safe Zone	Heist		1	2	0	belle	melodie	8-bit	darryl	colt	Byron	cordelius	clancy	kenji	angelo	Moe
4	Hard Rock Mine	Gem Grab		2	2	1	Lola	Surge	Darryl	Max	Kit	Frank	Moe	Kenji	Stu	Gene	Lily

Discussion

- ggpyparser seeks to be a step forward in supplying public esports data by providing an easy way to access data from Liquipedia, a community-run, open-access esports resource
- However, future work will entail increasing the ease of use of the package
  - Since ggpyparser uses names from Liquipedia for columns, some names will have shortened names with less transparent meanings
- In addition, as some Liquipedia pages are automatically generated, with the wikicode describing a call to an internal database, future work will involve expanding the package's ability to handle HTML requests in conjunction with wikicode requests

Acknowledgements

The author would like to thank Liquipedia for making this data publicly available, and the site's many contributors whose efforts made this project possible.