Nick Hingtgen

DeRon Holden

James Rucker

ETL Report

Magic the Gathering is a complicated trading card game with ample opportunities for data collection and analysis. The pauper format was chosen for its stability and its card pool restrictions, which lower the number of obscure gameplay mechanics that may interfere with data collection. Pauper tournament data in the form of decklists were scraped from [mtggoldfish](https://www.mtggoldfish.com/metagame/pauper#paper) (a popular site for players). The card names from these decklists were then used to scrape [gatherer](https://gatherer.wizards.com/Pages/Default.aspx) from Wizard’s of the Coast’s (the company that makes Magic) website. Splinter, BeautifulSoup, and Jupyter Notebook were used for scraping both websites. MongoDB was used for the database, making it non-relational.

Tournament data for the pauper format was scraped from mtggoldfish. Many of Magic’s formats have been recently plagued by bannings, which changes the card pool from which decks may be built. Different card pools lead to vastly different decks and mixing them may skew card selection. Therefore, the tournaments examined were limited to those following the most recent pauper banning. In addition, only the top 4 decks were scraped to ensure that the data better represented the metagame, and tacitly filtered out lucky but otherwise underperforming decks. The scraping was performed by navigating to the page with all pauper tournaments on it, then navigating to the desired tournament in an outer loop, and then the top 4 decks of each tournament by an inner loop. From the deck page, data was scraped for each card by name and number of copies. As a quirk, the deck page stores 3 separate identical decklists depending on how the cards might be acquired (Paper, Magic Online, and Magic Arena). Once all decklist data was collected, the number of all copies of each card were summed then divided by 3 (the listtriplicate decklists meant each card was triple-counted). A CSV of card names and counts was then saved for later use.

Scraping data from gatherer for each card required conceptualizing cards by card anatomy. After the first autofilled entry of the search function was selected, the name was verified to match that given before the scraping occurred. If it did not, the card name was added to a list so these values could be dropped and properly scraped later. Conceptualizing around card anatomy broke each card into distinct rows from which relevant code could be extracted. Once the relevant data were extracted, they were appended to their respective lists to be combined into a dataframe later. From here, the browser backed into gatherer and the process repeated. Once completed, cards whose scraped name did not match the one given were dropped from the dataframe. The urls for their respective gatherer pages were compiled manually (fewer than 10 cards needed this over all testing and the final run) and then used to navigate the browser by url, otherwise repeating the above process. The dataframe of previously mismatched cards was then appended to the original dataframe (whose original mismatched entries were dropped), leaving a complete dataframe with all important data for the cards used in pauper tournaments.

In MongoDB the “raw\_count” and “mtg\_merged” were imported as CSV files. From there the “mtg” database was created and within it 2 collections for each CSV was made. Each column within the CSV’s was read as string (text), Int32 (whole numbers) and Double (decimals). This was done so each column/row within the files would be read correctly without any errors. After that Pymonogo and Pandas were used to speak with MongoDB in order to display the desired data in Jupyter notebook.