EDH-LLM: Using NLP techniques to build card game decks

Theo Hui tchui@ucsd.edu
Linus Lin l6lin@ucsd.edu
Evelyn Yee eyee@ucsd.edu
Jingbo Shang jshang@ucsd.edu

Abstract

In this project, we propose a novel system for generating playable decks for the card game, "Magic: The Gathering," in the commander format (also known as EDH; Elder Dragon Highlander). This task combines hard and soft restrictions, maximizing the validity, playability, and power of a deck, in addition to an issue of scale, as there are thousands of possible cards, leading to over 10^{285} total possible decks. To address the holistic demands of this deck-building task in this large search space, our system features Natural Language Processing techniques, like Word2Vec, alongside a Large Language Model (LLM) in the loop. In this way, we extract the maximum utility from the semi-structured text data provided with each card and create valid decks with almost no supervised data. Alongside an existing heuristic for evaluating the power level of EDH decks, we also propose a new, count-based metric for predicting card synergy based on unlabeled historical play data.

Code: https://github.com/evelynyee/edh-llm

1 Introduction

Magic the Gathering is a deck-building card game where players collect cards and assemble custom decks to play against each other.

The Elder Dragon Highlander (EDH) format is loosely structured for deck-building; players must select a Commander card (see Figure 1a for an example) to lead 99 other cards in a 1v1v1v1 last-man-standing format. Building a successful deck necessitates the following considerations with varying levels of subjectivity:

1. Hard Restrictions:
    - The commander must be a Legendary Creature (more on keywords/typing later). There are thousands of Legendary Creature cards.
    - The other 99 cards in the deck must fit into the color typing of the commander.
2. Soft Restrictions:
    - Synergies: Cards in a deck must play well with each other. This can be expressed in many ways, including typal synergies, playstyle archetypes, and combos.
    - Power Curve: Decks must strike a balance between three main archetypes of cards; card draw, mana generation, and gameplan cards must be evenly balanced in a deck to allow a smooth gameplay experience. Missing a turn due to lack of cards or mana, or failing to execute a gameplan are all risks that can be minimized by good deck composition.
    - Rule Zero: Decks must be fun to play *with* **and** play *against*. This can be achieved by building a deck that is, on average, of comparable speed/strength to other decks in the playgroup.

Additionally, the sheer quantity of possible cards available for deck building causes the process of manually creating unique decks to become incredibly daunting. Naively searching through 27,000 cards to find 99 to suit your deck is impractical, and the evaluation of these subjective qualities is difficult to automate using plain heuristics. We are inspired by these challenges to research a way to efficiently generate legal decks that also meet the various soft restrictions of the EDH deckbuilding problem.

To address these diverse constraints, we frame the challenge of deck building as an information retrieval task, selecting the remaining cards for a commander like a search engine selects relevant documents for a query. Inspired by this framework, we propose a pipeline for leveraging numerical card data and card body text for synergy-informed deck building.

With our data consisting primarily of text, natural language processing (NLP) tools, such as Word2Vec (Mikolov et al. 2013) and ChatGPT, can help us find similarities between documents and significantly reduce the search space. Word2Vec transforms text into its vector representations, which will help us find similarities between documents, acting as a broad search for related cards. ChatGPT has significant potential to help us narrow down the documents even further, as it has been trained on several hundred gigabytes of text with billions of parameters, allowing us to implicitly access large embeddings with strategic prompting. ChatGPT will allow us to semi-automatically cluster documents together through the use of guided prompts. This can either be done through the use of iterative

(a) Sample Commander

(b) Extremely synergistic card

Figure 1: These cards share a high synergy despite having relatively different (tokenwise) oracle text.

prompts, asking ChatGPT to pick which of two documents is most similar to a cluster (triplet task), or simply iteratively prompting ChatGPT to the add single most similar document to the cluster.

2 Related Work

EDHRec is the current most popular deck analysis tool, generating card recommendations by scraping human-made deck data from deck build/publishing sites. It is able to recommend cards that are highly synergistic with a chosen commander via an algorithm similar to TF-IDF. This provides insight on cards that interact well with a specific commander vs. commander staples that are generically good. However, this site has major drawbacks:

- Many commanders can utilize multiple theme archetypes (ex. +1/+1 counters, tribal, token, aristocrat, blink). For a given commander, EDHRec will recommend the best cards out of each viable theme, however including all of these cards in a deck would result in an overall weak deck due to a lack of synergy within the other 99.
- Many commanders are used in "Pre-con" decks sold ready-built by Wizards of the Coast. Due to this, on deck sites, most decks including these commanders will have little/no modification. This gives the illusion that many weak cards are in fact synergistic due to their inclusion in this commander's decks.
- EDHRec is incapable of finding new synergies that have not already been found. For

example, when new cards are released, EDHRec is unable to provide the same level of recommendations as more established commanders for weeks.

From a more academic perspective, current research regarding deckbuilding systems is also not perfectly suited for this application. For example, Zhang et al. (2022); Kowalski and Miernik (2020) focus on small-scale deckbuilding (i.e. selecting cards from a set of 5 option), and Fancher (2015); Stiegler et al. (2016) implement heuristic-based search algorithms for assembling decks based on complex characteristics. This effort, though effective at test time, requires significant human supervision/labeling for training, and does not allow for the discovery of new synergies. In contrast, (Kowalski and Miernik 2020; Chen et al. 2018) use sequential processes, like the evolutionary algorithm, to assemble decks, which can be computationally intensive and slow at test-time. Historical systems also rely heavily on play data and do not leverage the information provided in the semi-structured oracle text of the card itself. We re-frame this process of deck-building as a text-based information retrieval task, where a user provides a seed card (i.e. their Commander), and the rest of the deck is built by searching through the text representations of the remaining ~27,000 cards for the most relevant supporting cards.

Information retrieval (IR) tasks are typically separated into two phases: using a fast method to select a mid-sized candidate pool from the large available search space, and then using a slower, more powerful neural system to re-rank this candidate pool for more fine-grained relevance. We aim to explore the use of Large Language Models (LLMs) for the re-ranking aspect of this task, taking advantage of these systems' ability to understand complex relationships in text. This application of LLMs for search tasks has been surveyed by Lin, Nogueira and Yates (2021), who provide a comprehensive view of applications of large transformer-based language models for single-stage and multi-stage text ranking for a variety of IR purposes. However, these methods are computationally intensive, as they require specialized training of millions of parameters over a large text corpus. To counter this need for custom LLM training, Sun et al. (2023) propose a pipeline for text ranking using large black-box LLM systems, like ChatGPT. They find significant performance improvements over other LLM approaches, including a passage re-ranking BERT model (Nogueira and Cho 2020) and a custom LLaMa model (Touvron et al. 2023). Inspired by these findings, we apply a ChatGPT-based approach to re-ranking in our specialized deckbuilding task.

3 Methods

For our deck-building tool, we present a 2-phase information-retrieval pipeline, using the Commander card as the search query. In the candidate pool selection phase, we embed each card into a high-dimensional vector representation and select the top 500 most similar cards according to cosine similarity. To generate each card's vector representation, we pass its oracle text through Word2Vec (Mikolov et al. 2013), an open-sourced text-based embedding model. We augment these text embeddings with other domain-specific, non-text features of the cards, like card type, color, and keywords.

After reducing the full list of ~27,000 available cards down to a candidate pool of 500, we

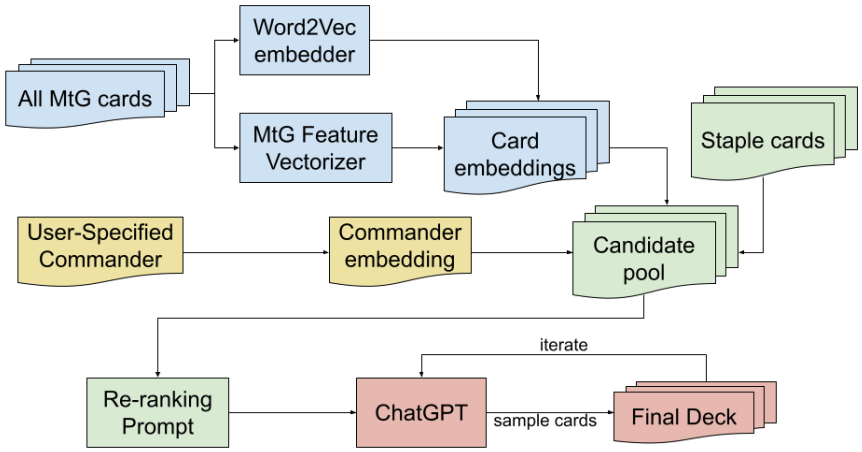


Figure 2: Our deckbuilding pipeline.

further bolster it by adding well-known, strong cards, commonly known as staples, to the pool. We then use ChatGPT, through the OpenAI API, to break ties and select the best 99 cards to form a competitive deck. We prompt it to iteratively select 5 cards at a time, to ensure that each subset of cards added synergizes well with all of the previously selected cards.

3.1 Evaluation

From the over 900 potential Commander cards, we randomly selected a set of 100 for validation/pipeline tuning and 189 for testing our final system. For each Commander, we will evaluate our generated deck against 3 baseline decks:

- **Greedy Popularity (EDHRec)**: Select the 63 cards which are most frequently played with the selected Commander, according to decklist data from EDHRec (See Section 2).
- **Embedding-only (Cos_sim)**: Gather the 63 cards with the most similar vector embeddings to the commander. Similar to directly sampling a candidate pool of size 63.
- **Random Selection (Manual_rand)**: From our final candidate pools (incorporating Word2Vec embeddings, manual features, and ”staple” cards), randomly select 63 cards, rather than using ChatGPT to perform re-ranking and selection.

36 basic land cards will also be added to these decks, adding up to a total of 99 cards (excluding the commander), as they are generally necessities to include in decks. We will evaluate each deck (our EDH-LLM decks, along with the 3 baselines) on the following criteria, related to the constraints described in Section 1:

Hard Restrictions: We will automatically assess the color typing of the cards in the deck, to see that they match the commander’s color typing, which is a hard restriction of decks in this play format.

Synergy Heuristics: We will use scraped deck data from EDHRec to estimate the synergy between a pair of cards through a Bayesian probability measure. We define the synergy heuristic for two cards a, b as follows:

$$\text{synergy}(a,b) = \log\left(\frac{freq(a,b)^2}{freq(a)freq(b)}\right)$$

where $freq$ denotes the frequency of a card, or pair of cards, occurring in a deck in the EDHRec corpus.

If cards have negative/no synergy, they should co-occur less frequently together than they do on their own ($P(A \cap B) < P(A)P(B)$). If they have neutral synergy, their occurrence should be independent (i.e. $P(A \cap B) = P(A)P(B)$). If they have positive synergy, they should co-occur more frequently than they do separately ($(P(A \cap B) > P(A)P(B)$). The logarithm and the squaring of the numerator both improve the spread of scores for more practical use and interpretation.

We report the **average synergy** over all pairs in a deck (4,950 pairs in a 100-card deck) as well as the **Commander synergy** between all non-commander cards and the commander (99 synergy scores).

Power Heuristic: We will use a formula, developed by Gavin (2020), to estimate the power level of any EDH deck:

$$\frac{2}{A} + \frac{\frac{D}{2} + T + \frac{R}{2}}{2} + \frac{I}{20} = P$$

where

- **A** denotes the average CMC (Converted Mana Cost) of the deck
- **D** denotes draw that either allows you to see 3 cards, or a permanent that gives you repeatable draw
- **T** denotes tutors with CMC of 4 or less that also find combo pieces and other win conditions
- **R** denotes ramp cards with CMC of 2 or less
- **I** denotes interaction, such as counterspells, targeted removal, board wipes, and stax
- **P** denotes the power level of the deck

This formula aims to minimize **A**, while maintaining a healthy balance of **D**, **T**, **R**, and **I**. Each variable also has an associated weight with it, conveying the relative importance of each variable when considering power.

Playability Evaluation: To assess the subjective aspects of deckbuilding, we intended to develop a deck evaluation questionnaire to rank the proposed deck on a variety of qualities, like power balance, win speed and fun-factor, aggregating the scores for each question to get a holistic playability score for the deck. However, due to the cost and level of domain expertise required for these human evaluations, we did not have time to perform this more holistic, subjective evaluation.

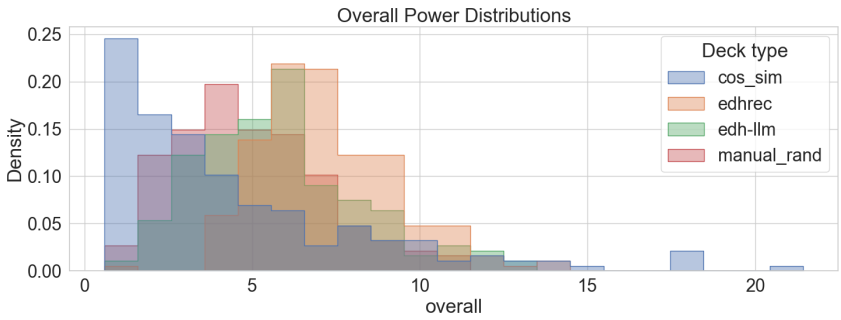


Figure 3: Calculated overall power level for each deck.

4 Results

4.1 Power Heuristic

In Figure 3, we observe that the overall power distribution for the Embedding-only baseline has a similar distribution to the other methods, but shifted left. This suggests that the Embedding-only deck-building method is worse at generating high power decks than the other methods. However, while the other methods have very similar distributions, the Greedy Popularity baseline decks seems to have the highest power, followed by our EDH-LLM decks, and the Random Selection decks. In other words, the historical play data seems to be the best indicator for generating a high power deck.

After that, our re-ranking system seems to improve on the custom candidate pools, which improve on the text embedding only candidate pools. This hierarchy of results seems to confirm the value of each step of our EDH-LLM pipeline, as each step improves on the previous results. One potential hypothesis for this incremental improvement is that each step of the pipeline incorporates more staple cards than the previous. Staples are so strong, that, in order to get the numerically highest-power deck, most of each deck could consist of just staple cards. From this perspective, it seems that the Greedy Popularity method's success may also be attributable to staple cards, because the play rates heavily favor the inclusion of staples, while the other methods include less staples, in favor of more synergistic cards.

Although the Greedy Popularity method resulted in the decks with the highest power levels, this method requires access to a large amount of historical play data, which provides a supervised signal for the viability of a deck. In contrast, the embedding-based baselines and our EDH-LLM system use almost no supervised data, only in the small list of staple cards for each color. This small amount of supervision also applies for every single commander, whereas the historical play data is distinct for each.

In Figure 4, we observe the each feature that makes up our power heuristic - cmc, draw, interaction, and ramp. The embedding-only baseline seems to have a tendency of going "all or nothing" when it comes to each of these features, meaning that it either contains an abundance of some features, while completely missing other features. The other methods

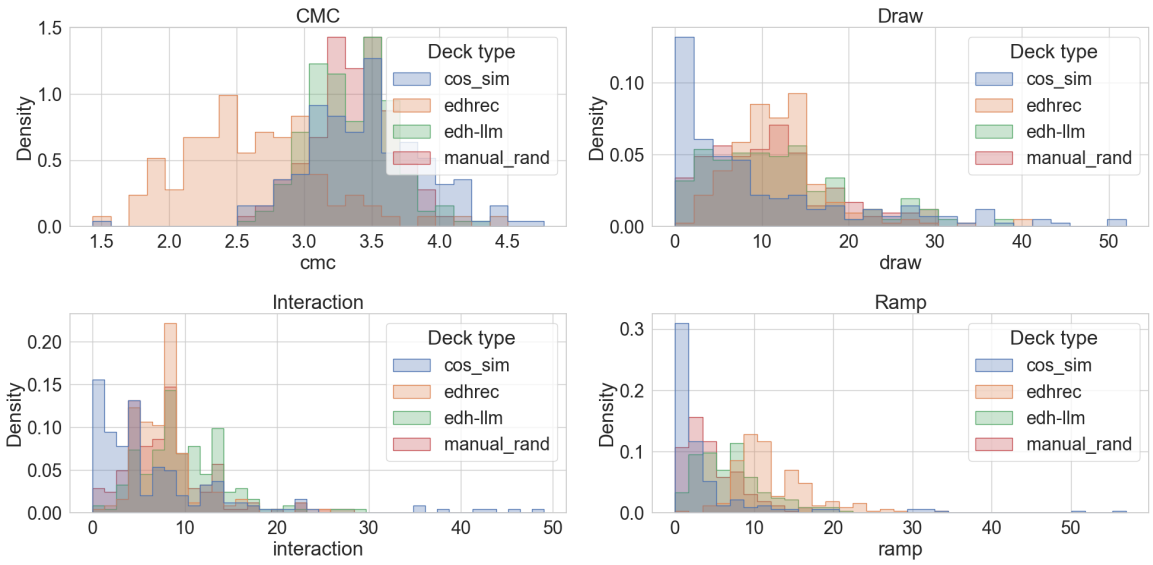
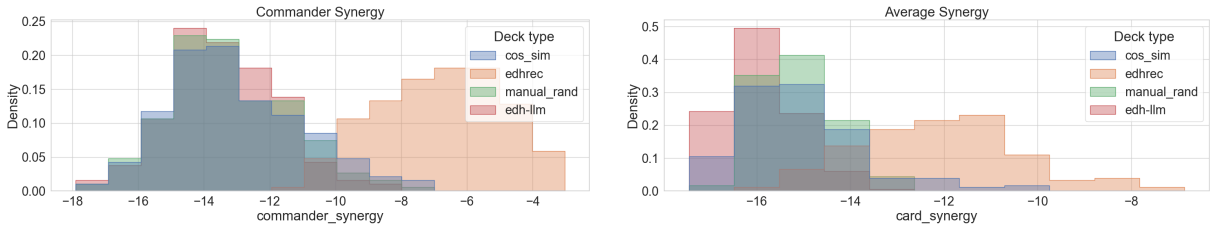


Figure 4: Power calculation components for each deck (see Section 3.1 for details)

have a generally similar distribution to each other, for all of the features.

4.2 Synergy Heuristic



(a) Commander synergy for each deck

(b) Average synergy for each deck

Figure 5: Synergy heuristic evaluation of each deck, with the 4 baseline methods and our system, EDH-LLM (see Section 3.1 for details). A higher score indicates that cards in the deck have been more frequently played together in the EDHREC (2024) database.

Our synergy heuristic evaluations can be seen in Figure 5. Because the Greedy Popularity baseline decks were built solely from the co-play frequency of each card with the commander, those decks have higher synergy scores than the rest, especially for the commander synergy heuristic (comparing synergy between the commander and each of the other cards). Looking at the commander synergy scores, the other three deck-building systems all seemed to perform similarly.

For the average synergy heuristic (averaging synergy across all pairs of cards, excluding the commander), the Greedy Popularity baseline decks again generally demonstrate the

highest synergy, but the distribution of average synergy is much more spread. The other two baselines, which were built using our card embedding system, seem to demonstrate similar average synergy to each other, and our EDH-LLM final decks seem to demonstrate slightly lower synergy. This difference in synergy may be explained by the greater diversity in card roles. In real life, many cards can become popular for reasons external to their actual play characteristics, due to outside factors, like availability and cost to buy the physical card. These outside effects may have created artifacts in the Greedy Popularity baseline deck and synergy heuristics that were not reflected in our other decks, which require much less outside data to build.

Figure 6: Marneus Calgar, a commander that heavily favors card draw and token creation but lacks ramp and interaction. Picking cards which are too similar leads to an imbalanced deck.

Our GPT algorithm (by design) aims to combine both synergistic cards with generic good-stuff 'staples.' These staples, while not the most synergistic, are necessary to balance out the needs of the deck. For example, "Marneus Calgar" (Figure 6) is a commander with strong card draw and synergizes with token creation. However, a successful deck still requires ramp and interaction cards, which in this case lower the synergy score but increase power scores. In general, our GPT model tended to pick these staples at a higher rate than our other two homebrewed baselines, explaining the lower average synergy but higher average power.

5 Conclusion

In this project, we developed a custom pipeline for suggesting decks for the EDH format of play with Magic the Gathering, incorporating NLP tools, like Word2Vec (Mikolov et al. 2013) and ChatGPT, with an information extraction approach to this under-studied domain. Decks created by our system achieves moderate power levels, only slightly lower

than historically played decks for a much smaller amount of supervised data.

Our results illustrate the complexity of the deck building task. For example, even the simplified evaluation of numerical power scores requires balancing a variety of constraints which arise from complex interaction. In fact, our deck evaluation does not even directly consider some more more advanced card types, like tutors or combos, which are common for high level play (cEDH). Further experimentation with the inclusion of these factors could potentially see GPT rise in power level.

Additionally, we wrote the GPT algorithm to accept a custom target power level but did not extensively test with different power levels, instead opting for a high power target to have a relevant comparison to our simpler baselines. Ultimately, the goal of building a deck is not always to reach the highest power level possible but rather create decks comparable to one's opponents, for the most fun in casual play. In the future, we hope to further explore this customization capability. To do this, we would perform more extensive testing, including a subjective, human evaluation, and engineer our system to cater to an individual player's needs, including their playstyle, target strength, and competitors.

References

Chen, Zhengxing, Chris Amato, Truong-Huy Nguyen, Seth Cooper, Yizhou Sun, and Magy Seif El-Nasr. 2018. "Q-DeckRec: A Fast Deck Recommendation System for Collectible Card Games."

EDHREC. 2024. "EDHREC." [Link]

Fancher, Will. 2015. , Sep. [Link]

Gavin. 2020. "My EDH Power level formula." Nov. [Link]

Kowalski, Jakub, and Radosław Miernik. 2020. "Evolutionary Approach to Collectible Card Game Arena Deckbuilding using Active Genes."

Lin, Jimmy, Rodrigo Nogueira, and Andrew Yates. 2021. "Pretrained Transformers for Text Ranking: BERT and Beyond."

Mikolov, Tomas, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. "Efficient Estimation of Word Representations in Vector Space."

Nogueira, Rodrigo, and Kyunghyun Cho. 2020. "Passage Re-ranking with BERT."

Stiegler, Andreas, Claudius Messerschmidt, Johannes Maucher, and Keshav Dahal. 2016. "Hearthstone deck-construction with a utility system." In *2016 10th International Conference on Software, Knowledge, Information Management Applications (SKIMA)*. [Link]

Sun, Weiwei, Lingyong Yan, Xinyu Ma, Shuaiqiang Wang, Pengjie Ren, Zhumin Chen, Dawei Yin, and Zhaochun Ren. 2023. "Is ChatGPT Good at Search? Investigating Large Language Models as Re-Ranking Agents."

Touvron, Hugo, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Au-

relien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. 2023. "LLaMA: Open and Efficient Foundation Language Models."

Zhang, Yulun, Matthew C. Fontaine, Amy K. Hoover, and Stefanos Nikolaidis. 2022. "Deep Surrogate Assisted MAP-Elites for Automated Hearthstone Deckbuilding."

Appendices

A ChatGPT Prompt

```
{"role": "system", "content": "You are building a commander deck for
magic the gathering. You will be given a pool of already selected cards,
a candidate pool, some heuristics of the cards already selected as well as
a target heuristic score. Select five the cards from the candidate pool that will
move the current heuristic score toward the target score.
For example, if the target ramp score is 10 and the current score is 3,
pick cards that will aid with mana generation.
Return simply the names of the cards you have selected,
separated by semi-colons. Do not deviate from the formatting specified."},
                    {"role": "user", "content":
                    "Selected: " + '; '.join(cur_deck)
                    "Candidate Pool: " + '; '.join(pool)
                    "Current Power: " + str(cur_power)
                    "Target Power: " + str(target_power)}
```

B Further Results

folder	cmc	draw	interaction	overall	ramp
cos_sim	3.452893	10.222222	7.222222	4.506141	3.994709
edhrec	2.620288	11.910053	8.322751	7.247856	12.243386
gpt	3.342160	11.804233	10.005291	5.819461	7.058201
manual_rand	3.354173	10.968254	8.031746	4.904552	4.640212

(a) Mean Power Scores

folder	cmc	draw	interaction	overall	ramp
cos_sim	3.454545	6.0	4.0	3.267797	1.0
edhrec	2.562500	11.0	8.0	7.022807	11.0
gpt	3.328125	10.0	10.0	5.618357	6.0
manual_rand	3.359375	10.0	7.0	4.662440	4.0

(a) Median Power Scores