Building a Simple Eternal Card Drafting AI Using NLP Techniques

Eternal is a digital collectible card game by Dire Wolf Digital similar to other games like Magic: The Gathering and Hearthstone. Players assemble decks of cards composed of creatures and spells with various abilities and face off in tactical one on one battles. Typically, players assemble their decks from their entire collection of cards but another game format called Draft is also very popular and this is what we will be focusing on today.

The Draft format forces players to build their decks one card at a time from a very limited pool of random cards. In Eternal, each player begins the draft by opening a pack of 12 random cards. They select one of these cards to add to their deck and pass the remaining 11 to the player on their left. They’ll then receive 11 cards from the player on their right, choose one for their deck and pass along the remaining 10. This process continues until the pack is exhausted and then repeated with new random packs three additional times. By the end of the draft, each player will have selected 48 cards from which they need to construct a 45 card deck. Players have unlimited access to basic Power cards (resource cards) when constructing their deck and typically about 15 are included, leaving about 30 cards from the draft. The Draft format challenges a player’s ability to construct a cohesive deck on the fly. Not only must the player commit to card choices without any guarantee of what cards they may have access to later, players must also interpret the draft meta of what types of decks other players are attempting to build based which cards aren’t being selected from the packs. This emphasizes simple, reliable, and flexible deck strategies built from relatively common cards. Once the player has completed their deck, they play against other draft players up to 7 wins or 3 losses (whichever comes first) at which point their draft run is complete and they must draft a new deck.

Our Dataset

Conveniently for us, some of the top Eternal draft players have been compiling a list of their most successful draft decks over the last ~6 months. As they play Draft, any deck which achieves a full 7 wins is added to the list in the form of a deck link to the community deckbuilding website [www.EternalWarcry.com](http://www.EternalWarcry.com). From these links, we can decode the exact cards included in each deck. The dataset can be found as several Excel files at this website under the OneDrive link:

http://farmingeternal.azurewebsites.net/

Using this dataset of successful draft decks we will first try to understand card selecting trends and what constitutes a good deck and then we will use this knowledge to design a simple AI that can intelligently draft a reasonable deck. One final note, the overall pool of cards available in the Draft format is regularly changed by the game developers every few months to keep the challenge fresh for players. The dataset above includes decks from several of these previous cycles, but we’ll be restricting our analysis to only the currently running Draft cycle called “The Flame of Xulta” which began just over a month ago in early October. Unfortunately, out of the thousands of deck lists this limits us to only using the ~140 most recent decks (although this amount should grow and improve our results over time until a new cycle begins and we must start over).

Analyze the Dataset

For this analysis, we’ll be using many techniques common in Natural Language Processing. This is a natural approach because our task requires us to build an understanding about the meaning and relationships of cards in an unsupervised manner. If we replace the term “card” with “word” and “deck” with “document” everything else is essentially exactly the same.

We start by building our dictionary of every card included in the dataset and assign each a sequential numeric ID. We’ll use these IDs to convert each deck list into a card count vector and use the overall dataset to build a TF-IDF matrix. This matrix tells us how common a card in a deck is chosen relative to how commonly it’s selected overall. Since Sigils (basic power cards) are not included in the draft, we remove them from the deck lists so they don’t influence our analysis. With the TF-IDF matrix, we’ll use Non-negative Matrix Factorization (NMF) to extract common “topics” or deck trends. NMF is a dimension reduction technique based on co-occurrence similarities between cards. With some manual tuning, I settled on 20 NMF components as providing good results. In NLP, NMF is often used for extracting shared topics from document datasets but in our case the equivalent is deck types. Each NMF component captures the importance of every card in our dictionary to a particular deck type. In the table below, I’ve broken down each NMF component based on the percentage of cards it contains from the different color factions and we can quickly see how different components focus on individual factions or combinations of factions. It’s important to remember that decks aren’t exclusively defined by a single component but typically a combination of several related components.

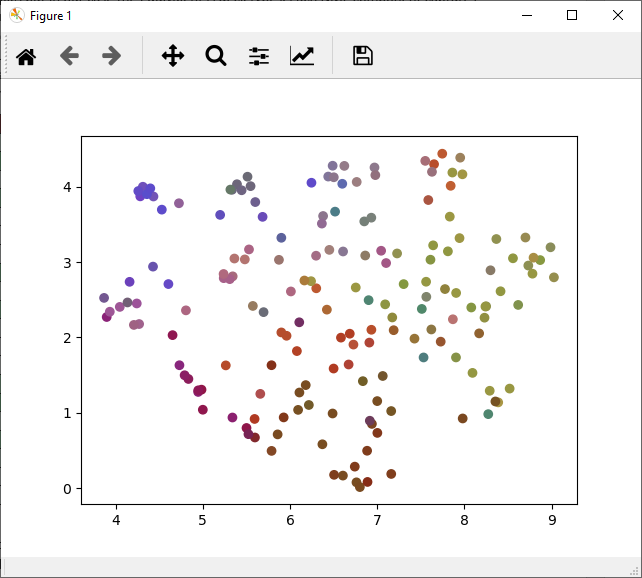
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Comp# | Fire | Time | Justice | Shadow | Primal |
| 1 | 0.16 | 0.24 | 0.50 | 0.05 | 0.05 |
| 2 | 0.55 | 0.04 | 0.37 | 0.02 | 0.01 |
| 3 | 0.10 | 0.20 | 0.09 | 0.53 | 0.09 |
| 4 | 0.45 | 0.41 | 0.09 | 0.03 | 0.02 |
| 5 | 0.03 | 0.09 | 0.07 | 0.42 | 0.39 |
| 6 | 0.02 | 0.45 | 0.09 | 0.41 | 0.02 |
| 7 | 0.03 | 0.43 | 0.43 | 0.06 | 0.05 |
| 8 | 0.51 | 0.07 | 0.09 | 0.32 | 0.01 |
| 9 | 0.14 | 0.29 | 0.48 | 0.02 | 0.07 |
| 10 | 0.46 | 0.07 | 0.43 | 0.03 | 0.01 |
| 11 | 0.07 | 0.39 | 0.10 | 0.39 | 0.06 |
| 12 | 0.03 | 0.34 | 0.11 | 0.09 | 0.43 |
| 13 | 0.03 | 0.40 | 0.45 | 0.07 | 0.04 |
| 14 | 0.39 | 0.37 | 0.10 | 0.09 | 0.06 |
| 15 | 0.54 | 0.14 | 0.17 | 0.09 | 0.05 |
| 16 | 0.15 | 0.49 | 0.19 | 0.10 | 0.07 |
| 17 | 0.46 | 0.08 | 0.39 | 0.03 | 0.04 |
| 18 | 0.17 | 0.52 | 0.17 | 0.09 | 0.07 |
| 19 | 0.07 | 0.39 | 0.42 | 0.05 | 0.06 |
| 20 | 0.06 | 0.08 | 0.38 | 0.43 | 0.05 |

Most of the components capture faction pairs which are quite commonly played. Interestingly, each faction has an exclusive component (1, 3, 15, 18) except for Primal/Blue which may suggest that faction is relatively weak in the current cycle and unable to form the core of a deck on its own. Also note how Time/Yellow is significantly represented in 11 different components compared with 2 for Primal/Blue.

Since each component encodes the importance of every card in our dictionary for a particular deck type, we look at which are the most important cards. It’s important to note here that while I say “important” really what I mean is “most commonly included” in decks of that type. For our purpose, we’ll assume these are equivalent but they may not be. Below I’ve included links to the most important cards for each NMF component. If you’re familiar with Eternal, we can quickly see that all of these generally make sense and that many of the more powerful cards in the current cycle have been appropriately identified. In the table above several components seemed to overlap with very similar faction colors (the Time/Yellow and Shadow/Purple combination for example), but when we look at the actual cards we can see that each component focuses on slightly different strategies.

|  |  |
| --- | --- |
| Comp# | Deck Link |
| 1 | <https://eternalwarcry.com/deck-builder?main=7-62:1;7-70:1;7-78:1;7-65:1;7-64:1;2-70:1;7-87:1;7-173:1;7-85:1;7-80:1;7-79:1> |
| 2 | <https://eternalwarcry.com/deck-builder?main=7-157:1;7-22:1;6-5:1;7-5:1;7-65:1;7-6:1;7-23:1;5-7:1;5-10:1;1-42:1;7-73:1> |
| 3 | <https://eternalwarcry.com/deck-builder?main=6-213:1;7-137:1;7-127:1;7-145:1;7-126:1;7-131:1;1-265:1;1-266:1;7-134:1;7-147:1;4-236:1;4-220:1;7-142:1;4-216:1;1-270:1;6-189:1> |
| 4 | <https://eternalwarcry.com/deck-builder?main=7-154:1;2-175:1;6-34:1;7-26:1;7-7:1;7-50:1;4-26:1;1-96:1;7-48:1;1-102:1> |
| 5 | <https://eternalwarcry.com/deck-builder?main=7-203:1;7-109:1;7-96:1;7-136:1;6-136:1;7-118:1;7-137:1;5-231:1;7-125:1;7-120:1;7-202:1;7-100:1;5-120:1;4-197:1;7-105:1> |
| 6 | <https://eternalwarcry.com/deck-builder?main=7-185:1;5-222:1;7-129:1;7-126:1;1-278:1;7-134:1;7-188:1;6-46:1;7-143:1;7-56:1;2-35:1;6-69:1;4-60:1;7-45:1;1-117:1;2-149:1;1-254:1;7-33:1;7-51:1;7-140:1;5-181:1;7-125:1;4-238:1;4-64:1;7-184:1;7-46:1;7-55:1> |
| 7 | <https://eternalwarcry.com/deck-builder?main=7-174:1;7-87:1;1-133:1;7-43:1;7-49:1;4-57:1;4-101:1;5-246:1;1-80:1> |
| 8 | <https://eternalwarcry.com/deck-builder?main=7-17:1;7-8:1;7-16:1;7-14:1;7-145:1;7-168:1;7-130:1;1-392:1;7-170:1;7-26:1;7-12:1;4-22:1;5-194:1;7-7:1;7-27:1;7-5:1;7-25:1> |
| 9 | <https://eternalwarcry.com/deck-builder?main=7-73:1;7-63:1;4-133:1;7-89:1;1-114:1;1-80:1;6-118:1;7-64:1;2-73:1;4-101:1;2-80:1;6-64:1> |
| 10 | <https://eternalwarcry.com/deck-builder?main=5-78:1;1-38:1;5-7:1;7-157:1;7-14:1;7-79:1;7-78:1;7-75:1;5-14:1;4-142:1;3-18:1;7-25:1;6-26:1;7-20:1;6-97:1;5-81:1;7-23:1;5-87:1> |
| 11 | <https://eternalwarcry.com/deck-builder?main=7-57:1;7-140:1;7-38:1;6-219:1;1-396:1;4-204:1;7-123:1;1-302:1;3-220:1;6-226:1;4-53:1;7-31:1;5-181:1;3-212:1;5-42:1;1-264:1;6-210:1;1-265:1;1-503:1;7-48:1;7-43:1;7-145:1;5-252:1;7-126:1;7-127:1> |
| 12 | <https://eternalwarcry.com/deck-builder?main=7-181:1;7-113:1;7-102:1;7-98:1;7-101:1;7-114:1;7-96:1;7-106:1;7-118:1;4-57:1;1-197:1;1-220:1;1-203:1;1-212:1;1-209:1;5-252:1;6-138:1;7-95:1;7-178:1;3-154:1> |
| 13 | <https://eternalwarcry.com/deck-builder?main=7-86:1;7-43:1;7-39:1;4-135:1;7-175:1;1-505:1;7-78:1;5-108:1;7-64:1;3-65:1;7-46:1;7-51:1;6-91:1;6-52:1;1-141:1;2-64:1;1-150:1;7-70:1;7-45:1;0-21:1;0-58:1;7-68:1;7-40:1> |
| 14 | <https://eternalwarcry.com/deck-builder?main=4-12:1;7-40:1;7-1:1;1-76:1;3-30:1;6-21:1;4-27:1;6-78:1;1-55:1;5-241:1;7-36:1;3-63:1;7-45:1;7-38:1;7-56:1;6-5:1;2-175:1;7-23:1;7-51:1;1-75:1;7-14:1;7-22:1;7-9:1;7-7:1;7-26:1;7-154:1> |
| 15 | <https://eternalwarcry.com/deck-builder?main=7-9:1;7-6:1;1-31:1;1-17:1;7-25:1;7-22:1;5-10:1;7-28:1;7-7:1;7-3:1;6-7:1> |
| 16 | <https://eternalwarcry.com/deck-builder?main=7-56:1;7-49:1;7-43:1;7-37:1;7-173:1;2-50:1;4-65:1;7-51:1;4-63:1;1-75:1;6-7:1;4-64:1;7-152:1> |
| 17 | <https://eternalwarcry.com/deck-builder?main=7-14:1;2-70:1;4-14:1;5-196:1;7-75:1;7-9:1;3-123:1;7-62:1;7-86:1;6-5:1;1-31:1;7-6:1;7-63:1> |
| 18 | <https://eternalwarcry.com/deck-builder?main=6-82:1;7-55:1;1-80:1;7-40:1;6-46:1;7-48:1;7-51:1;7-151:1;1-160:1;1-93:1;7-57:1;7-49:1;1-69:1;7-58:1;6-66:1> |
| 19 | <https://eternalwarcry.com/deck-builder?main=7-40:1;7-72:1;7-46:1;7-38:1;7-173:1;7-78:1;0-58:1;7-82:1;1-69:1;6-118:1;7-73:1;5-91:1;7-56:1;1-137:1;2-70:1;4-113:1;6-61:1;6-230:1;6-104:1;7-74:1> |
| 20 | <https://eternalwarcry.com/deck-builder?main=7-197:1;7-134:1;7-62:1;7-129:1;6-97:1;1-159:1;7-147:1;5-236:1;7-127:1;1-130:1;7-198:1;6-189:1;7-86:1;6-201:1;5-87:1;3-274:1;5-252:1;1-171:1> |

We can even use UMAP to plot the decks in our dataset in 2D space to get a visual feel for how these different deck types relate to each other. In the plot below, each point represents a deck in our dataset and is colored according to the relative mix of factions it includes.



Define Good Decks

Our analysis so far has broken down the current Draft meta and given us a lot of insight into how players are prioritizing cards for their decks but how can we leverage this knowledge to design an AI which can effectively draft its own deck. The key assumption we’ll make to help us on our way is that these NMF components represent card distributions for ideal decks and the closer we can match these ideal distributions the better our deck will be. This is a big assumption we’re making and it’s pretty easy to prove how this assumption falls apart in practice but for our immediate needs it’ll mostly get us to where we’re trying to go.

Remember that actual decks are not defined by a single component but usually a combination of several so we need to understand what actual decks look like in component space. Converting our entire deck dataset into component space, we can use a technique like HDBSCAN to extract clusters. With some manual parameter tuning, we can identify 10 unique clusters and we can take the median of decks in that cluster to define the cluster center. I should note here that the number of decks we have in the dataset currently isn’t really enough to extract good quality clusters but for now we’ll just press on.

Build the AI

Now given a deck list, we have everything we need to appropriately characterize that deck in a meaningful way. An AI, however, requires an optimization metric to guide its decisions. I mentioned before that we can think of our components as probability distributions across our card dictionary. Multiplying our cluster matrix by our component matrix gives us the distribution for our specific deck types. If we can guide our AI to match one of these deck distributions it should be able to put together a reasonably cohesive deck. Fortunately, there’s a fairly simple metric for comparing probability distributions called KL Divergence. So given our current deck list which our AI is drafting, we can compare it to an ideal deck list and quantify the difference between the two with a score of zero meaning a perfect match.

At this point we have all the tools we need to piece together our AI. At each step in the draft, our AI has its current deck list and is presented with a set of random cards and it must choose a card based on what it’s previously chosen. First we identify which type of deck the AI is building by assigning it to one of the deck clusters based on Euclidean distance to the cluster centers in component space. This gives us the ideal card distribution for that type of deck, and now we can calculate how much each card choice minimizes the KL Divergence between the current deck and the ideal. The AI can then choose the appropriate card in a greedy fashion and move to the next step.

Once the AI has drafted 48 cards, it then needs to downselect its final deck of 30. In the exact same way as before, we’ll evaluate each card in the deck calculate which card best minimizes the total KL Divergence when removed. We’ll repeat this process in a stepwise fashion until only 30 cards remain. Finally, we’ll add 15 Sigils to the deck based on a simple proportionality rule to bring us to our final deck size of 45.

Results

Since we know the cards which are available in the draft pool and their relative rarities, we can code up a simple function to sample from the card pool to build the random packs of 12 cards. In this case we ignore Legendary rarity cards because they’re too rare in our dataset to model appropriately. With this we can set up 12 AIs to draft against each other (where each AI chooses a card from a randomly generated pack and passes the remainder to the next AI). I’ve posted links below to the decks that the AIs drafted. Many of these actually look like pretty reasonable decks although some decks clearly failed to converge to something meaningful. Most also feature a pretty good power curve and mix of units and spells. Overall fairly impressive results for such a simple approach.

|  |  |
| --- | --- |
| Deck# | Deck Link |
| 1 | <https://eternalwarcry.com/deck-builder?main=1-126:6;1-249:6;1-63:1;1-276:1;5-172:1;1-187:2;5-196:1;5-78:1;1-171:1;5-246:1;1-504:1;1-141:1;3-241:1;6-136:2;6-189:1;0-58:1;7-127:1;7-134:1;7-62:1;7-87:1;7-137:1;7-145:1;7-136:1;7-65:1;7-86:1;7-197:2;7-77:1;4-238:1;7-204:1;7-100:1;7-121:1;7-135:1> |
| 2 | <https://eternalwarcry.com/deck-builder?main=1-249:7;1-63:1;1-1:7;5-5:1;6-189:1;6-5:1;6-230:1;0-60:1;7-127:2;7-147:1;4-12:1;1-266:1;7-16:2;7-48:1;7-136:1;7-25:1;7-130:2;7-7:1;7-8:1;0-35:1;0-36:1;7-17:1;1-302:1;7-5:1;1-55:1;1-12:1;2-156:1;7-29:1;7-170:1;7-169:1> |
| 3 | <https://eternalwarcry.com/deck-builder?main=4-133:2;1-126:7;1-63:5;1-187:1;1-1:1;4-101:1;5-97:1;5-115:1;1-75:1;2-80:2;1-141:1;6-56:1;6-118:1;7-40:1;7-79:1;4-12:1;7-64:1;7-39:1;7-57:2;7-73:3;1-212:1;7-55:1;7-174:1;1-80:1;7-50:1;7-76:1;1-102:1;7-15:1;7-4:1> |
| 4 | <https://eternalwarcry.com/deck-builder?main=1-126:2;1-249:3;1-63:4;1-96:1;1-187:4;5-41:1;1-1:3;1-31:1;5-47:1;5-123:1;6-46:1;1-407:1;6-5:1;6-230:1;0-60:1;7-124:1;7-40:1;7-134:1;7-62:1;7-6:1;7-43:1;7-154:1;7-145:1;7-138:1;7-64:1;7-105:1;7-181:1;7-118:1;7-81:1;7-123:1;7-26:1;7-95:1;3-167:1;2-194:1;7-92:1> |
| 5 | <https://eternalwarcry.com/deck-builder?main=1-249:7;1-63:6;1-187:2;5-231:1;1-218:1;4-213:1;6-213:1;3-212:1;6-181:1;6-189:1;1-503:1;4-236:1;1-369:1;7-33:1;4-53:1;7-129:1;7-51:1;7-142:1;7-56:2;7-147:1;7-43:1;7-140:1;7-145:1;7-28:1;7-48:1;7-138:1;2-211:1;7-39:1;7-49:1;7-130:1;7-104:1;1-72:1> |
| 6 | <https://eternalwarcry.com/deck-builder?main=1-126:6;1-249:7;1-187:2;5-87:1;1-171:1;1-504:1;1-141:1;3-145:1;6-136:1;6-182:1;6-190:1;7-129:2;7-134:1;7-143:1;7-147:1;7-211:1;1-266:1;7-78:1;7-86:1;7-77:1;7-81:1;7-195:1;7-14:1;7-73:1;1-270:1;7-61:1;3-202:1;3-239:1;7-203:1;7-120:1;7-204:1;7-135:1> |
| 7 | <https://eternalwarcry.com/deck-builder?main=1-126:7;1-63:7;1-146:1;5-78:1;5-131:1;1-75:1;4-111:1;4-65:1;2-55:1;6-46:1;6-104:1;6-52:1;6-71:1;1-168:1;1-333:1;7-33:1;7-40:1;7-43:1;7-87:1;7-63:1;7-49:1;7-68:1;7-70:2;7-174:4;0-11:1;7-74:1;7-85:1;1-102:1> |
| 8 | <https://eternalwarcry.com/deck-builder?main=5-84:1;1-126:6;1-1:9;5-7:1;1-142:1;5-10:1;1-504:1;1-37:1;1-32:1;6-91:1;6-5:1;6-31:1;1-131:1;1-15:1;2-70:1;7-80:1;7-211:1;7-16:1;7-65:2;7-72:1;7-78:1;7-14:1;7-20:1;7-23:1;4-27:1;7-22:3;7-5:1;1-55:1;7-3:1> |
| 9 | <https://eternalwarcry.com/deck-builder?main=1-63:4;1-187:11;5-120:1;1-197:1;3-154:1;3-63:1;1-357:1;4-173:1;6-155:1;6-46:1;1-407:1;7-105:2;7-181:2;7-101:1;7-102:1;7-49:1;7-113:1;7-96:1;7-98:1;7-179:1;1-68:1;7-178:1;7-114:1;7-92:1;7-104:1;7-111:1;7-100:1;7-108:1;3-106:1;3-157:1> |
| 10 | <https://eternalwarcry.com/deck-builder?main=1-126:1;1-249:2;1-63:6;4-216:1;1-187:6;1-193:1;5-241:1;5-91:1;5-134:1;3-63:1;1-407:1;7-37:1;7-51:1;1-69:1;7-131:1;7-48:1;7-138:1;7-105:1;7-181:1;7-101:1;7-118:1;1-100:2;7-55:1;7-179:1;1-415:1;7-36:1;0-33:1;7-178:1;7-104:1;7-111:1;3-171:1;1-360:1;3-106:1> |
| 11 | <https://eternalwarcry.com/deck-builder?main=1-126:1;1-249:2;1-63:1;1-187:1;1-1:10;1-193:1;1-31:1;1-321:1;3-63:1;6-21:1;6-1:1;2-28:1;7-62:1;7-6:2;4-12:1;7-28:2;7-25:1;7-7:2;0-36:1;4-27:1;7-17:1;7-22:1;1-55:1;7-125:1;1-17:1;7-170:1;0-3:1;2-194:1;0-6:1;7-203:1;1-72:1;7-4:1> |
| 12 | <https://eternalwarcry.com/deck-builder?main=2-68:1;1-126:2;1-249:2;1-63:5;1-187:5;1-1:1;5-241:1;3-154:1;1-336:1;1-213:1;6-46:1;6-91:1;6-137:1;6-149:1;6-150:1;7-124:1;4-64:1;7-51:1;7-43:1;7-24:1;7-131:1;7-138:1;7-35:1;7-49:1;7-59:1;1-80:1;7-5:1;7-95:1;7-76:1;1-196:1;7-183:1;1-208:1;7-92:1;7-111:1;3-106:1> |

Next Steps

We made a number of assumptions over the course of this analysis, in particular that our NMF components characterized the card distribution for an ideal deck but this is actually not a very good assumption. These components only really capture how frequently a card was chosen relative to the overall dataset. This assumption does manage to hold together because it’s clear that certain cards are objectively better than others and are therefore chosen more frequently and since the dataset is already downsampled to only those decks which achieved a full 7 wins we can assume that these cards were chosen deliberately by experienced players. At a surface level, however, this doesn’t account for the different rarities of cards in the draft packs (certain cards turn up far more frequently than others). Nor does it attempt to understand the role which different cards play in an overall deck and how some cards may be functionally interchangeable with others.

What we need is a better understanding of the individual cards and how they contribute to the functional pieces which come together to make a good deck. For this we’ll need to take a step up in sophistication and use the techniques of word embedding. Instead of translating our cards into an NMF component space, we’ll use word embedding to learn a new latent space which can capture a far more nuanced understanding of individual cards and their relationships to each other. This will give us a window into a card’s practical function in a deck rather than just its relative co-occurrence and should allow our AI to make better decisions.