Natural Language Processing

Mitchell Brogan

9/5/2022

**Abstract:** The goal of this project was to create a content-based recommender that would be able to take a magic card’s name as input and output the top ten most similar cards. An application of this tool would be to provide deck-builders with options for what to include in their deck. A popular format of the game known as Commander (aka EDH) has a deck building restriction which prohibits multiples of any given card. This tool would allow players to find other cards with similar abilities for them to maintain the playstyle they want whilst adhering to the format’s restrictions. In addition, some cards may be priced outside of a player’s budget, and they could use this tool to find a more affordable alternative for their desired card. A trade study was conducted to determine the most relevant recommender for this task.

**Design:** Prior to tokenizing the card text, I had dropped unnecessary rows, removed unwanted non-alphanumeric characters, removed a card’s name from its rules text if it were to reference itself, as well as other rudimentary cleaning techniques. Next, I applied a whitespace tokenizer to isolate individual words as well as unique magic related text (ex. {t}). I then used spacy to lemmatize and remove stop-words in an effort to reduce dimensionality. I then performed a tfidf vectorizer to weight all of the terms based on how frequently the terms appear in the corpus. PCA was applied to the output of the tfidf vectorizer to reduce the dimensionality by finding the lowest number of components that yielded an explained variance ratio greater than 0.70. Finally, the pairwise distances between the input card and all other cards being considered are taken, and the top ten closest cards are presented to the user.

The aforementioned approach was the foundation that was applied to investigating NMF and KMeans clustering to determine the best model.

**Data**: 26,000+ rows, with 70+ columns. The only column of interest was oracle\_text which contained 0-100+ words per card. This dataset was oracle\_cards.json downloaded from <https://scryfall.com/docs/api/bulk-data>.

**Algorithms**: Each of the three models (TFIDF, NMF, KMeans) was run using single word tokens/two-gram tokens, as well as cosine/Euclidean distance between points. Each combination of these amounted to twelve different models tested against each other to determine the best recommender.

**Tools**:

* Pandas
* Numpy
* Regex
* Spacy
* Nltk
* sklearn (PCA, KMeans, NMF, pairwise\_distances)
* scattertext

**Communication**: In addition to the powerpoint presented, the Jupyter Notebooks will be posted on the following GitHub link: https://github.com/MitchellB9/Magic-Card-Recommender