# Magic Machine Learning

Jakob, Anton, Jonathhan and Jens

ECKMASCEP



What is a Magic The Gathering?

- Trading Card Game
- Released 1993 by Wizards of the Coast
- Created by Richard Garfield





#### What is a Magic Card?



#### What is a Magic Card?

- Mana cost
- Туре
- Rarity



#### What is a Magic Card?

- Mana cost
- Туре
- Rarity
- Name
- Art
- Subtype
- Rules Text
- Power/Toughness

(Release year, set, flavor text, artist)





## Playing with data

#### Data

- 103.976 rows x 79 columns
- 103.976 cards x 79 features
- Some cards appear multiple times
  - Good when working with art
  - Less good when working with text/balancing

	-	
0	-	
0	 	
0		
4		
Ò		
~	_	

79

#### Data reduction

#### Rows:

- Prototypes
- Joke cards
- "Outliers"

#### Columns:

- Hard to work with:
  - Artist
  - Printings
- Very few entries:
  - hasContentWarning (29 vs. 103.947)
  - relatedCards (58 vs. 103.947)



## One-hot encoding

Card #	Туре	
1	Artifact, Creature	
2	Enchantment	

Card #	Artifact	Creature	Enchantment
1	1	1	0
2	0	0	1



#### **Boosted Decision Tree on original\* dataset**



## Predicting color from art

### Multi-label classification



32 Combinations of color

### **Color and Color Identity**















## Convolution - Resnet



## **Boosted decision tree**

## Generating power, toughness and subtype



#### Predicting power



#### Predicting toughness





## Text generation

### Models

Model	Strength	Weakness
Character model	Can spell new or rare words	Less context understanding
Token model	Understand full words/phrases	Struggles with rare tokens

Token model with character fallback(hybrid model)

- Token model = heavy lifting
- Character model fills any "holes" the token model leaves behind
- Token model: "Draw <unk> cards."
- Character model: "three"

#### Card generation

<|color|> Blue <|cost|> {1}{U}{U} <|rarity|> Common <|type|> Sorcery

<|endofheader|>

Return target creature card from your graveyard to your hand.

<|endofcard|>



### Training

Char Model Loss per Validation Step 5 epochs ---- Val Char Loss ---- Train Char Loss 5000 iteration pr. epoch 4 \_ Around 2.5 hours 3 \_ Loss 2 1 5000 10000 15000 20000 25000 0 iteration (Global Step) Token Model Loss per Validation Step --- Val Token Loss ---- Train Token Loss 8 6 055 2 al for the second of the second and the second s 0 5000 10000 15000 20000 25000 iteration (Global Step)

#### Results

- <|color|> Green <|cost|> {1}{G} <|rarity|> Common <|type|> Instant <|endofheader|> Prevent all combat damage that would be dealt this turn.
- <|color|> Black <|cost|> {6}{B}{B} <|rarity|> Uncommon <|type|> Sorcery <|endofheader|> Destroy all creatures. They can't be regenerated.
- <|color|> Red <|cost|> {2}{R}{R} <|rarity|> Rare <|type|> Creature
   <|endofheader|> Flying Whenever this creature attacks or blocks,
   exile the top card of your library
- <|color|> Colorless <|cost|> {5} <|rarity|> Mythic <|type|> Artifact <|endofheader|> Creatures you control get + 2/ + 2. Whenever you tap a permanent for {C}, add an additional {C}. Whenever you cast a colorless spell, you gain 2 life

## Art generation



https://en.wikipedia.org/wiki/Variational\_autoencoder



2k

4k

6k



## **Optimization**

Wandb Sweep

#### Hyperparameter Importance





### Reconstruction

Latent space 8

Latent space 256

Original

## Reconstruction 100 epoch







50 epoch



500 epoch





Drawing from a gaussian distribution





































## **Conditional Sampling**

#### **Prompt Based Conditioning**

Manacost: 4 Color: Blue Rarity: Mythic Type: Creature







https://www.researchgate.net/figure/The-forward-and-backward-processes-of-the-diffusion-model-The-credit-of-the-used-images\_fig1\_382128283

## **Diffusion Optimization**

#### Pretrained model

Finetuning - 10 Epochs (18 hours)

Hyperparameter Optimization

Conditioning



## **Diffusion Results** 200 steps 100 steps 50 steps Text based conditioning **Green Elephant Red Goblin Blue White Sphinx** Prompt based conditioning

# A whole new release!

#### Generator script

#### Input:

- Color
- Mana value
- Туре
- Rarity

#### Output:

- Text (inkl. keywords) from: Self-trained GPT-2 model
- Power, toughness, subtype from: XGBoost
- Art from: Arcane Diffusion model
- Name from: Pretrained GPT-2 model

## \*placeholder\*

## Appendix

## Data inspection and cleaning

Violin Plot Matrix of creatures





#### Magic: The Gathering Color Combinations Distribution

#### CNN Resnet50 hyperparameters

```
Freezing layer 1-3:
```

```
max_epochs =25
```

```
learningrate = 0.001
```

```
No frozen layers:
```

```
max_epochs =25
```

```
learningrate = 0.0001
```

#### More CNN predictions



# XGBoost for predicting power, toughness and subtypes

#### Power

	precision	recall	f1-score	support
				a da Vina kar
0.0	0.82	0.46	0.59	414
1.0	0.73	0.53	0.62	1570
2.0	0.56	0.78	0.65	2142
3.0	0.55	0.49	0.52	1336
4.0	0.54	0.51	0.52	796
5.0	0.64	0.54	0.58	519
6.0	0.60	0.60	0.60	329
7.0	0.58	0.63	0.60	108
8.0	0.52	0.58	0.55	60
9.0	0.73	0.57	0.64	28
10.0	0.43	0.50	0.46	12
11.0	1.00	0.86	0.92	7
12.0	1.00	0.76	0.87	17
13.0	1.00	0.20	0.33	5
accuracy			0.60	7343
macro avg	0.69	0.57	0.60	7343
weighted avg	0.62	0.60	0.59	7343



#### Toughness

	precision	recall	f1-score	support
0.0	0.75	0.86	0.80	133
1.0	0.75	0.80	0.77	1726
2.0	0.68	0.73	0.70	1783
3.0	0.65	0.59	0.62	1464
4.0	0.67	0.67	0.67	1024
5.0	0.77	0.68	0.72	589
6.0	0.80	0.76	0.78	354
7.0	0.84	0.81	0.82	126
8.0	0.91	0.87	0.89	70
9.0	0.90	1.00	0.95	28
10.0	0.85	0.79	0.81	14
11.0	1.00	1.00	1.00	8
12.0	1.00	1.00	1.00	16
13.0	0.89	1.00	0.94	8
accuracy			0.71	7343
macro avg	0.82	0.82	0.82	7343
weighted avg	0.71	0.71	0.71	7343



### Subtype

#### Hamming Loss: 0.00658863789979021 Subset Accuracy (Exact Match): 0.3531254255753779 Jaccard Score (micro): 0.3541888461223487

#### Hyperparameter optimization

# Random integers between 50-499
# Random floats from 0.01 to 0.30 (0.01 + 0.29)
# Random integers between 3-9
# Random floats from 0.6 to 1.0 (0.6 + 0.4)
# Random floats from 0.6 to 1.0 (0.6 + 0.4)

#### # Random search

```
random_search_power = RandomizedSearchCV(
    estimator=xgb,
    param_distributions=param_dist,
    n_iter=20,
    scoring='accuracy',
    cv=3,
    verbose=2,
    random_state=42,
    # n_jobs=-1
```

## Card text generation

#### Transformer

#### Embedding

- Turn tokens into vectors(numbers)

#### Positional encoding

- We tell the model the order words

#### Self attention

- Learns which words to pay attention to in a sequence
- fx. "Tap target creature. That creature doesn't untap during its controller's next untap step."
- Each token gets 3 vectors: query(what i'm looking for), key(what I contain) and value(info I carry)

Multi-head attention: more attention = better understanding

Feed forward network

- 2 layer MLP
- helps combine features nonlinearly

#### Text generation parameters

- 5 epochs
- 5000 iteration pr. epoch
- 64 batch size
- 90% train
- 10% validation
- Ir = 0.000746
- temp = 0.8
- 16 heads
- 10 layers