ECE1786 Project Final Report "A.I. Meet Yu-Gi-Oh!"

Word Count: 1965, excluding qualitative examples and footnotes
Penalty: 0
Jianyu Wen 1002839690
Yuanzhuo Wang 1003149416

Permissions

Jianyu Wen

- Permission to post video: yes
- Permission to post final report: yes
- Permission to post source code: no

Yuanzhuo Wang

- Permission to post video: yes
- Permission to post final report: yes
- Permission to post source code: no

Introduction

"The Yu-Gi-Oh! Trading Card Game" ("Yu-Gi-Oh!") is a world-famous trading card game (TCG). While officially designed by KONAMI, players have also tried to write new "Yu-Gi-Oh!" cards to play with. We found Machine Learning (ML) a very appropriate tool for this task, as transformer-based text generation have recently shown great success in generating similar outputs such as stories.[1] However, while there exist prior advances in text generation in other TCGs, there are not as many notable researches available on "Yu-Gi-Oh!". Therefore, we are interested in exploring Natural Language Processing's (NLP) ability to generate card text in the style of "Yu-Gi-Oh!". We coined our project with the name "A.I. Meet Yu-Gi-Oh!".

Figure 1: Example of an official "Yu-Gi-Oh!" card, "A.I. Meet You". [2]

For our project, we use the card name as the pipeline input to generate every data needed to print a fully-functional "Yu-Gi-Oh!" card. The user have the ability to specify the card type, but we have also designed a classifier model in the pipeline that can detect what card type suits the input name best before passing the input to the text generator.
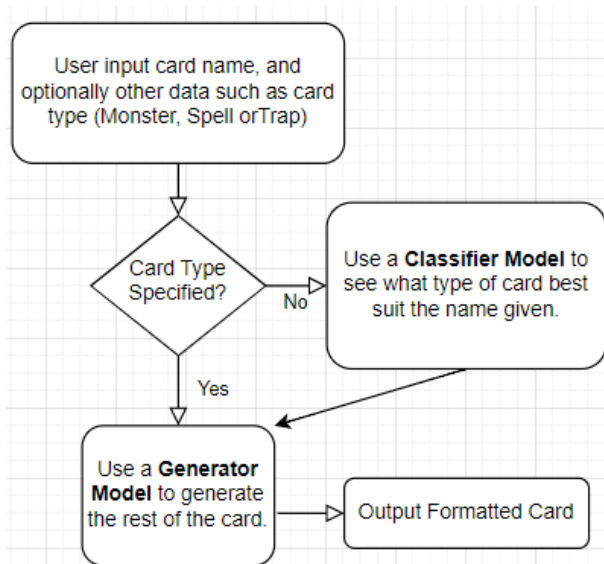
Figure 2: Simplified Flowchart of our Pipeline

Background

Card text generation is similar to other forms of text generation tasks, such as story generation. However, our task requires a stricter styling requirement, as modern English "Yu-Gi-Oh!" cards are written in an official set of grammar rules[1], the generator must learn this style. Another difference is that TCGs have cards designed to work with each other. Since both are special challenges to our task, it was difficult to find any paper within the team's technical level. However, we did find related works that we were able to learn from. The "minimaxir/magic-the-gathering" model on Huggingface is capable of generating "Magic: The Gathering" (MTG) card text with GPT-2-Simple.[3] It encodes the various data on a MTG card into different text fields by separating them between specialized tokens, and the model is capable of formatting the card in a predictable schema. This work was especially influential in our data processing. On the other hand, we found social media content in regards to "Yu-Gi-Oh!"-specific text generation. A Twitter account regularly posts cards generated by their model trained using GPT-2-Simple. [4] Another video on Youtube also demonstrated a game of "Yu-Gi-Oh!" using all A.I. generated cards. [5] Since the displayed cards are cherry-picked, we cannot tell how well their models have worked in general. However, we were able to confirm that our task is possible using a model similar to GPT-2.

---

[1] "Yu-Gi-Oh!" cards use a set of official grammar rules named "Problem-Solving-Card-Text" (PSCT). New cards are deliberately designed in PSCT to promote speedy understanding.

Data Gathering & Processing

To construct our dataset, we used the "Yu-Gi-Oh! API" made by YGOProDeck. [6] The API accepts queries with filters, so we used a Python request to gather all officially released cards post-2012[2], and retrieved a giant JSON response. Each piece of card data is given in a nested dictionary consisting of various data fields according to card typings.

There are 3 major card types in "Yu-Gi-Oh!": Monster Cards, Spell Cards, and Trap Cards. While Spell Cards and Trap Cards are similar in their formatting[3], Monster Cards have data that are specific to its sub-type. As data processing, we first removed cards irrelevant to our task[4]. Using our knowledge in "Yu-Gi-Oh!", we trimmed the dataset leaving only data crucial to game mechanics. Learning from the MTG model, we also added labels to separate texts into different data fields. Examples can be found in the Appendix 1. In total we processed 6864 cards, including 4544 Monster cards, 1349 Spell cards, and 971 Trap cards. They are placed in 3 separate files for fine-tuning.

Since there was a significant imbalance in card types, we applied data augmentation for Spell/Traps for training the classifier. We used the Python library "nlpaug", which has a "ContextualWordEmbsAug" function that uses a pretrained BERT model to performance synonym replacement or random insertion based on word embeddings calculation.[7], [8] "Synonym Replacement" randomly chooses words and replace them with random synonyms, whereas "Random Insertion" inserts synonyms of a word.[8] Figure 4 shows the implementations of both the insert and the substitute methods. Since the card names are short, we add or change only 1 single word.

```
import nlpaug.augmenter.word as naw
```

Figure 3: Import nlpaug Library

```
aug_insert = naw.ContextualWordEmbsAug(
    model_path='bert-base-uncased', action="insert",aug_min=1, aug_max=1,device='cuda')
aug_sub = naw.ContextualWordEmbsAug(
    model_path='bert-base-uncased', action="substitute",aug_min=1, aug_max=1,device='cuda')
augmented_text = aug_sub.augment("Pot of Greed")
print(augmented_text[0])
```

Figure 4: Substitute and Insert Augmentation Implementation

---

[2] We only use modern cards as dataset because they are all written in PSCT, while the others may not.
[3] Spell and Trap cards only have 3 text fields: the card name, the card effect, and the sub-type.
[4] Normal monsters do not have card effects, but have flavour texts instead, which does not use PSCT, which we are training our model to learn. Skill cards are exclusive to the Speed Duel format of "Yu-Gi-Oh!" and thus out of scope, but they were included in the JSON.

The program generates new names by randomly choosing entries from the Spell/Traps set and having half of them using the insert method and another half using the substitute method, until both Spell/Traps each have 4544 names. Table 1 shows some examples of the results.

Name	Insertion	Substitution
Lighting Storm	Violent Lighting Storm	Winter Storm
Dark Ruler No More	Dark Lord Ruler No More	Ancient Ruler No More
Evenly Matched	Evenly Well Matched	Evenly Results

Table 1: Data Augmentation Examples

Architecture and Software

As mentioned, our pipeline consists of a classifier model, and 3 text generator models. The classifier model determines what card type best fits the given name if no card type is given. It is a Cohere model that is fine-tuned on the Cohere website by uploading data in csv format shown in Figure 5.

	A	B
1	name	type
2	A.I. Challenge You	Trap Card
3	A.I. Shadow	Trap Card
4	A.I.'s Show	Trap Card
5	A.I.Q	Trap Card
6	Abominable Chambe	Trap Card
7	Abyss Actors Back Sta	Trap Card

Figure 5: Dataset for the classifier

Generators for each card type were initially produced using the free CohereXL pre-trained model by submitting the fine-tune datasets as txt files. Examples of the datasets can be found in Appendix 1, but in addition we used a custom separator "—SEPARATOR—" to separate different cards. The fine-tuning took around 2 hours per model. Currently Cohere's model parameters are not shared publicly. Figure 6 illustrates fine-tuning on Cohere.
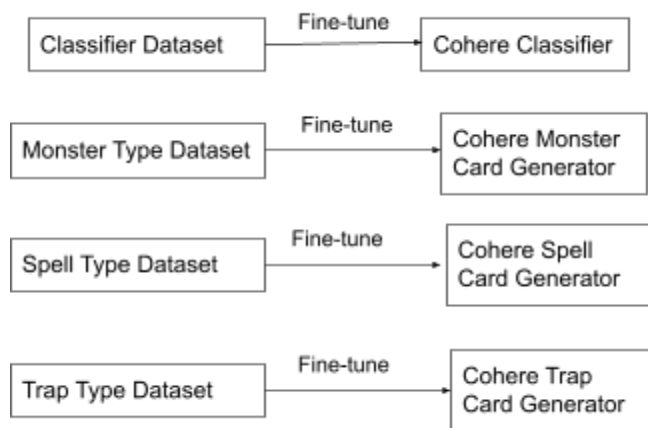
Figure 6: Cohere Fine-tuning

We also explored GPT-3 for text generations. Before passing to GPT-3, the datasets needed to be converted into the JSONL format by OpenAI with the prompt being the card names and the completions being the rest, as shown in Figure 7.

```
1   {"prompt":"<card><name>A.I. Challenge You<\\name\\>","completion":"<type>Trap Card<\\type\\><race>Continuous<\\race\\><desc>Monsters you
2   {"prompt":"<card><name>A.I. Shadow<\\name\\>","completion":"<type>Trap Card<\\type\\><race>Continuous<\\race\\><desc>Activate this card
3   {"prompt":"<card><name>A.I.'s Show<\\name\\>","completion":"<type>Trap Card<\\type\\><race>Normal<\\race\\><desc>While you control a Lin
4   {"prompt":"<card><name>A.I.Q<\\name\\>","completion":"<type>Trap Card<\\type\\><race>Continuous<\\race\\><desc>Activate only if you cont
```

Figure 7: GPT-3 JSONL

Generators for each type using GPT-3-Curie/Babbage were fine-tuned with uploaded JSONL files. Each model took 20 minutes to fine-tune on Babbage and 50 minutes on Curie on average. Figure 8 illustrates fine-tuning on GPT-3. Babbage and Curies have 1.3 billion and 6.7 billion parameters respectively. [9]
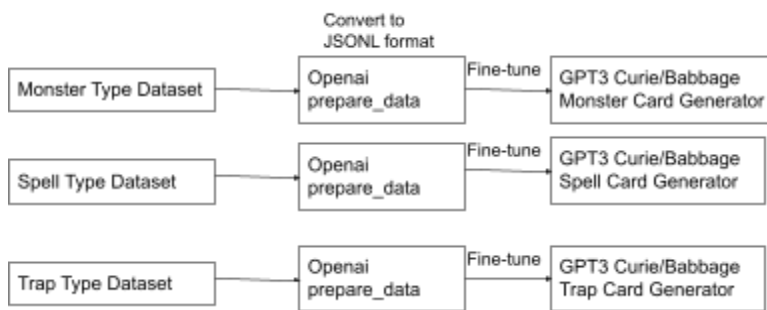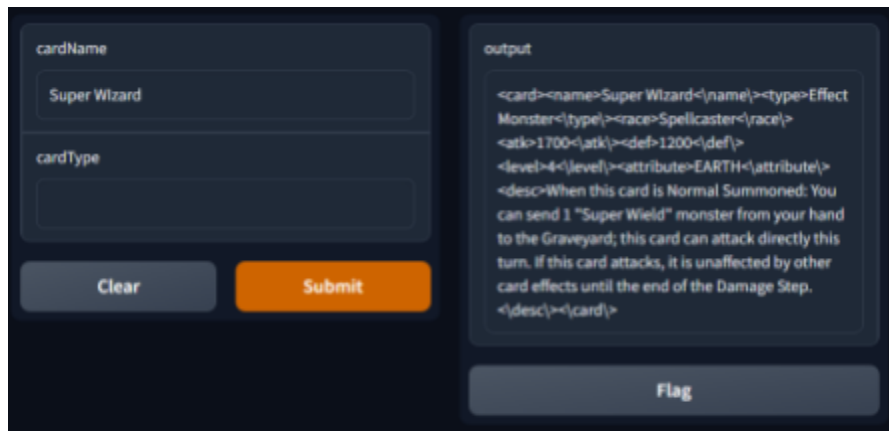

Figure 8: GPT-3 Fine-tuning

Figure 9: Gradio UI Output

All fine-tuned Cohere and GPT-3 models can be used in Python by giving API keys with model names. Based on the card type from the classifier or user, the generator chooses the matched typing from the possible card types to complete the card data. Output can be either displayed in Gradio UI or uploaded online for visualization. [10]

Figure 10: Output Visualization[10]

Quantitative Results

Cohere classifier achieved 82.5% validation accuracy, which is reasonable considering Spell/Trap names are difficult to distinguish.

Our team came up with a rubric for human evaluation for experienced players to hand-label with. The rubric contains 3 parts that are important for a card generation, as shown in Table 2. The "Writing Style" section has the highest weight as we believe this is the most fundamental measurement of quality.

Section	Mark
Writing Style • Does it make sense in game mechanics? • Is the card effect logical? Unambiguous? • Does it follow Yu-Gi-Oh! Grammar (PSCT)?	1 Mark, 50% Weight
Uniqueness • Is the effect interesting? Unique?	1 Mark, 25% Weight
Legitimacy • Does it sound like an official card? • Is the card thematic to its theme(if any)?	1 Mark, 25% Weight
Total Score	Weighted Sum

Table 2: Evaluation Rubric

We have generated 51 samples each from Cohere, GPT-3-Babbage and Curie using same inputs, temperature and top_p. We hand-labelled each result following our rubric, and results are shown in Table 3.

Model	Average Writing Style	Average Uniqueness	Average Legitimacy	Average Total Score
Cohere	0.647	**0.745**	0.431	0.618
GPT-3-Babbage	0.804	0.549	0.569	0.681
GPT-3-Curie	**0.824**	0.510	**0.725**	**0.721**

Table 3: Evaluation Results

GPT-3-Curie has the best quality outcomes overall, with its score being highest in total score, writing style, and legitimacy. However, our measurements are not without flaws. For example, small grammar mistakes are hard to evaluate. A carbon copy of an existing card can get a 0.75 score in our criteria, which we also did not account for. Regardless, we still think our rubric is the a very feasible and enforcable human measurement for a large amount of results.

All scoring data are available on the GitHub repository.

Qualitative Results

Even though our results generated by GPT-3 are noticeably higher in text quality in general compared to other models, there are quite a few interesting examples on both ends of the quality spectrum that are worth discussing. Here is an example of a extremely good card generation:

<card><name>Lyrilusc - Angelite Cockatiel<\name\><type>Effect Monster<\type\><race>Winged Beast<\race\><atk>100<\atk\><def>100<\def\><level>1<\level\><attribute>WIND<\attribute\><desc>During your Main Phase: You can Fusion Summon 1 "Lyrilusc" Fusion Monster from your Extra Deck, using monsters from your hand or field as Fusion Material. If a monster(s) you control is sent to the GY (except during the Damage Step): You can target 1 "Lyrilusc" monster in your GY, except "Lyrilusc - Angelite Cockatiel"; Special Summon it. You can only use each effect of "Lyrilusc - Angelite Cockatiel" once per turn.<\desc\><\card\>

The card is thematic[5] to its name with both its card values[6] and card effects[7].

For card values, The model learned this fact through the examples given in fine-tuning. In card text generations we use a lower temperature (0.4) and top_p (0.7) to encourage the model to give results like this. Only GPT-3 Curie is consistently capable of achieving thematically correct card values.

Thematic effects are much more difficult to generate since it requires the model to understand how similar cards are played and what is possible with the current card pool. We think this is currently too hard to do consistently.

Here are examples where the text quality is not so great:

<card><name>Rare Cat<\name\><type>Effect Monster<\type\><race>Beast<\race\><atk>800<\atk\><def>800<\def\><level>2<\level\><attribute>EARTH<\attribute\><desc>If this card is Normal or Special Summoned: You can add 1 "Rampdragon" monster from your Deck to your hand. You can banish this card from your GY;

---

[5] The extent of thematicity is a measurement of how it closely references the card name's theme or "archetype".
[6] Values here means ATK, DEF, Type, Attribute. A typical "Lyrilusc" monster in "Yu-Gi-Oh!" is a low ATK/DEF, WIND Winged-Beast effect monste, which is all true in this case.
[7] The first effect "Fusion Summon" a "Lyrilusc" Fusion monster, which is usable since there is exactly 1 existing monster this effect can "Fusion Summon" in "Yu-Gi-Oh!". The second effect can also work with the first effect, since a "Fusion Summon" performed by this card would also have "monster(s) you control" "sent to the GY", and thus trigger the second effect.

> Special Summon 1 "Rampdragon" monster from your Deck in face-down Defense Position. You can only use each effect of "Rare Cat" once per turn.<\desc\><\card\>

This card randomly refers to a "Rampdragon" monster, which does not exist in "Yu-Gi-Oh!", nor relate with the name "Rare Cat". This is a common occurrence, as seen also on this card generation:

> <card><name>"A"-Cell Scatterburst<\name\><type>Spell Card<\type\><race>Normal<\race\><desc>Target 1 =="A"-Cell==, =="Fusilade"==, and/or =="Gemini"== monster in your Graveyard; add them to your hand. If this card is in your Graveyard: You can shuffle 1 monster from your Graveyard into the Deck, then target 1 "A"-Cell, "Fusilade", and/or "Gemini" monster in your Graveyard; add it to your hand. You can only use 1 ==""A"-Cell Scatterburst"== effect per turn, and only once that turn.<\desc\><\card\>

This card also has trouble with its usage of the quotation marks[8]. This reflects that the model confuses sub-types of monsters with a name reference.

Occasionally, GPT-3 Curie copies entire card effects from an existing card onto a new one. This is not problematic until it becomes a carbon copy of an official card:

> <card><name>Seal of the War<\name\><type>Spell Card<\type\><race>Normal<\race\><desc>Reveal 1 card from your Deck and place it on top of your Deck, then your opponent reveals 1 card from their Deck and places it on top of their Deck. You can only activate 1 "Seal of the War" per turn.<\desc\><\card\>

While an official card "Top Share" has this effect: [11]

> Reveal 1 card from your Deck and place it on top of your Deck, then your opponent reveals 1 card from their Deck and places it on top of their Deck. You can only activate 1 "Top Share" per turn.

The generated card is a exact copy except its name. We think this is related to how GPT-3 is known to "memorize" text used in training.

In conclusion, our fine-tuned GPT-3 Curie model often has quirks that lowers quality, but it is sometimes capable of generating astoundingly good results.

---

[8] A Gemini monster is almost never referred with quotation marks in card text since it normally refers to a sub-type of monsters, rather than the name of a card. In addition, ""A"-Cell" is never correctly mentioned in this card text.

Discussion and Learnings

As some parts of the result discussions are already discussed, we conclude that overall GPT-3-Curie archived very good results, but the other two models (Cohere, and GPT-3-Babbage) are not far off. The model learned "Yu-Gi-Oh!" grammar well and can generate cards that are frequently very similar to official cards. We noticed that interesting and unique cards are more likely to have styling errors, meanwhile boring cards have few errors, since they copy existing cards or combine a few different cards. This is a problem since we require the results to be unique without sacrificing quality. We are unsure of how this can be fixed in training, and it is difficult to catch such "plagerisms" during result evaluations.

We also learned that GPT-3 is very strong on learning patterns. By reducing the data size, we could have a shorter fine-tining time, and we might be able to afford using GPT-3-Davinci.

Regarding the text generation tasks in ML, we realized that the most efforts are spent on data processing and result labelling. These steps currently rely on human input with no machine alternatives. We were fortunate to have spent time in the ratio of workload this time, but we could make better schedule plans should there be a similar ML task.

Individual Contributions

Jianyu Wen	Yuanzhuo Wang
Wrote Python Code to collect and process data from Yu-Gi-Oh! API	Wrote Python code to process data format and augmented data for classification
Fine-tuned text generation models on Cohere	Fine-tuned classification model on Cohere
Wrote code to connect the classifier model with 3 generation models	Converted and edited data into JSON format for GPT-3
Wrote a Gradio UI, help fine-tune GPT-3-Babbage and Curie	Fine-tuned GPT-3-Babbage and Curie for 3 generation models
Wrote Python code to run multiple inputs and output all results in tsv files	Wrote Python code to convert output results into JSONs that can be uploaded to a website for visualization
Evaluated and scored Cohere, GPT-3-Curie results, reviewed GPT-3-Babbage result scores	Evaluated and scored GPT-3-Babbage results

Table 4: Contribution Table

Appendix 1: Examples of Data Processing

Original Card Image	JSON from Yu-Gi-Oh! API	Dataset Entry
	{"data":[{"id":3611830,"name":"Endymion, the Mighty Master of Magic","type":"Pendulum Effect Monster","desc":"[Pendulum Effect] You can remove 6 Spell Counters from your field; Special Summon this card from the Pendulum Zone, then count the number of cards you control that can have a Spell Counter, destroy up to that many cards on the field, and if you do, place Spell Counters on this card equal to the number of cards destroyed. You can only use this effect of \"Endymion, the Mighty Master of Magic\" once per turn.\n[Monster Effect] Once per turn, when a Spell/Trap Card or effect is activated (Quick Effect): You can return 1 card you control with a Spell Counter to the hand, and if you do, negate the activation, and if you do that, destroy it. Then, you can place the same number of Spell Counters on this card that the returned card had. While this card has a Spell Counter, your opponent cannot target it with card effects, also it cannot be destroyed by your opponent's card effects. When this card	<card><name>Endymion, the Mighty Master of Magic<\name\><type>Pendulum Effect Monster<\type\><race>Spellcaster<\race\><atk>2800<\atk\><def>1700<\def\><level>7<\level\><attribute>DARK<\attribute\><scale>8<\scale\><desc>[Pendulum Effect] You can remove 6 Spell Counters from your field; Special Summon this card from the Pendulum Zone, then count the number of cards you control that can have a Spell Counter, destroy up to that many cards on the field, and if you do, place Spell Counters on this card equal to the number of cards destroyed. You can only use this effect of "Endymion, the Mighty Master of Magic" once per turn. [Monster Effect] Once per turn, when a Spell/Trap Card or effect is activated (Quick Effect): You can return 1 card you control with a Spell Counter to the hand, and if you do, negate the activation, and if you do that, destroy it. Then, you can place the same number of Spell Counters on this card that the returned card had. While this card has

	with a Spell Counter is destroyed by battle: You can add 1 Normal Spell from your Deck to your hand.","atk":2800,"def":1700,"level":7,"race":"Spellcaster","attribute":"DARK","archetype":"Endymion","scale":8,"card_sets":[{"set_name":"Structure Deck: Order of the Spellcasters","set_code":"SR08-EN001","set_rarity":"Ultra Rare","set_rarity_code":"(UR)","set_price":"3.11"}],"card_images":[{"id":3611830,"image_url":"https://images.ygoprodeck.com/images/cards/3611830.jpg","image_url_small":"https://images.ygoprodeck.com/images/cards_small/3611830.jpg"}],"card_prices":[{"cardmarket_price":"0.48","tcgplayer_price":"1.22","ebay_price":"2.35","amazon_price":"2.83","coolstuffinc_price":"1.99"}]}]}	a Spell Counter, your opponent cannot target it with card effects, also it cannot be destroyed by your opponent's card effects. When this card with a Spell Counter is destroyed by battle: You can add 1 Normal Spell from your Deck to your hand.<\desc\><\card\>
	{"data":[{"id":27381364,"name":"Spright Elf","type":"Link Monster","desc":"2 monsters, including a Level/Rank/Link 2 monster\r\nCannot be used as Link Material the turn it is Link Summoned. Your opponent cannot target monsters this card points to with card effects. During the Main Phase (Quick Effect): You can target 1 Level 2 monster in your GY, or, if your opponent controls a monster, you can target 1 Rank/Link 2 monster instead; Special Summon it. You can	<card><name>Spright Elf<\name\><type>Link Monster<\type\><race>Thunder<\race\><atk>1400<\atk\><attribute>FIRE<\attribute\><linkval>2<\linkval\><linkmarkers>['Bottom-Left', 'Bottom-Right']<\linkmarkers\><desc>2 monsters, including a Level/Rank/Link 2 monster Cannot be used as Link Material the turn it is Link Summoned. Your opponent cannot target monsters this card points to with card effects. During the Main Phase (Quick Effect): You

	only use this effect of \"Spright Elf\" once per turn.","atk":1400,"race":"Thunder","attribute":"FIRE","archetype":"Spright","linkval":2,"linkmarkers":["Bottom-Left","Bottom-Right"]],"card_sets":[{"set_name":"Power of the Elements","set_code":"POTE-EN049","set_rarity":"Ultra Rare","set_rarity_code":"(UR)","set_price":"11.34"}],"card_images":[{"id":27381364,"image_url":"https://images.ygoprodeck.com/images/cards/27381364.jpg","image_url_small":"https://images.ygoprodeck.com/images/cards_small/27381364.jpg"}],"card_prices":[{"cardmarket_price":"19.97","tcgplayer_price":"18.35","ebay_price":"9.46","amazon_price":"9.79","coolstuffinc_price":"15.99"}]}]}}	can target 1 Level 2 monster in your GY, or, if your opponent controls a monster, you can target 1 Rank/Link 2 monster instead; Special Summon it. You can only use this effect of "Spright Elf" once per turn.<\desc\><\card\>
	{"data":[{"id":72283691,"name":"Golden Castle of Stromberg","type":"Spell Card","desc":"Once per turn, during your Standby Phase, you must banish 10 cards from the top of your Deck face-down (this is not optional), or this card is destroyed. During your Main Phase: You can Special Summon 1 monster from your Deck that specifically lists the card \"Golden Castle of Stromberg\" in its text. You cannot Normal Summon/Set the turn you activate this effect (even if this card leaves the field). You can only use this effect of \"Golden Castle of Stromberg\" once per turn.	<card><name>Golden Castle of Stromberg<\name\><type>Spell Card<\type\><race>Field<\race\><desc>Once per turn, during your Standby Phase, you must banish 10 cards from the top of your Deck face-down (this is not optional), or this card is destroyed. During your Main Phase: You can Special Summon 1 monster from your Deck that specifically lists the card "Golden Castle of Stromberg" in its text. You cannot Normal Summon/Set the turn you activate this effect (even if this card leaves the field). You can only use

	When an opponent's monster declares an attack: Destroy the attacking monster, and if you do, inflict damage to your opponent equal to half the ATK that monster had on the field.","race":"Field","archetype":"Golden Castle of Stromberg","card_sets":[{"set_name":"Battles of Legend: Relentless Revenge","set_code":"BLRR-EN010","set_rarity":"Secret Rare","set_rarity_code":"(ScR)","set_price":"9.42"},{"set_name":"Duel Overload","set_code":"DUOV-EN094","set_rarity":"Ultra Rare","set_rarity_code":"(UR)","set_price":"1.99"}],"card_images":[{"id":72283691,"image_url":"https://images.ygoprodeck.com/images/cards/72283691.jpg","image_url_small":"https://images.ygoprodeck.com/images/cards_small/72283691.jpg"}],"card_prices":[{"cardmarket_price":"0.23","tcgplayer_price":"0.78","ebay_price":"1.93","amazon_price":"2.30","coolstuffinc_price":"6.99"}]}]}	this effect of "Golden Castle of Stromberg" once per turn. When an opponent's monster declares an attack: Destroy the attacking monster, and if you do, inflict damage to your opponent equal to half the ATK that monster had on the field.<\desc\><\card\>

References

[1] A. Vaswani *et al.*, "Attention Is All You Need," Jun. 2017, doi: 10.48550/arXiv.1706.03762. [Online]. Available: http://dx.doi.org/10.48550/arXiv.1706.03762. [Accessed: Dec. 09, 2022]

[2] "A.I. Meet You," *Yugipedia - Yu-Gi-Oh! Wiki*. [Online]. Available: https://yugipedia.com/wiki/A.I._Meet_You. [Accessed: Dec. 09, 2022]

[3] "minimaxir/magic-the-gathering · Hugging Face." [Online]. Available: https://huggingface.co/minimaxir/magic-the-gathering. [Accessed: Dec. 09, 2022]

[4] "YugiohbutAI," *Twitter*. [Online]. Available: https://twitter.com/YugiohbutAI. [Accessed: Dec. 09, 2022]

[5] *Attempting to Play w/ AI Generated Cards feat. @Hypnocorn*. (Oct. 12, 2022) [Online]. Available: https://www.youtube.com/watch?v=RfR3jjGAGkU. [Accessed: Dec. 09, 2022]

[6] "Yu-Gi-Oh! API Guide - Yu-Gi-Oh! Card Database," *YGOPRODeck*. [Online]. Available: https://ygoprodeck.com/api/. [Accessed: Dec. 09, 2022]

[7] "nlpaug.augmenter.word.context_word_embs — nlpaug 1.1.11 documentation." [Online]. Available: https://nlpaug.readthedocs.io/en/latest/augmenter/word/context_word_embs.html#. [Accessed: Dec. 11, 2022]

[8] E. S. Shahul, "Data Augmentation in NLP: Best Practices From a Kaggle Master," *neptune.ai*, Jul. 21, 2022. [Online]. Available: https://neptune.ai/blog/data-augmentation-nlp. [Accessed: Dec. 11, 2022]

[9] L. Gao, "On the Sizes of OpenAI API Models," *EleutherAI Blog*, May 24, 2021. [Online]. Available: https://blog.eleuther.ai/gpt3-model-sizes/. [Accessed: Dec. 11, 2022]

[10] "Yu-Gi-Oh! Card Maker." [Online]. Available: https://ygopro.org/yugioh-card-maker/. [Accessed: Dec. 12, 2022]

[11] "Top Share," *Yu-Gi-Oh! CARD DATABASE*. [Online]. Available: https://www.db.yugioh-card.com/yugiohdb/card_search.action?ope=2&cid=16869&request_locale=en. [Accessed: Dec. 10, 2022]