Proceedings of the 9th Games and Natural Language Processing Workshop @LREC2022, pages 54–60
Marseille, 25 June 2022

Generating Descriptive and Rules-Adhering Spells for Dungeons & Dragons Fifth Edition

Pax Newman, Yudong Liu
Western Washington University
516 High Street, Bellingham, WA, USA
{newmanp, liuy2}@wwu.edu

Abstract

We examine the task of generating unique content for the spell system of the tabletop role-playing game *Dungeons and Dragons Fifth Edition* using several generative language models. Due to the descriptive nature of the game, it presents a number of interesting avenues for generation and analysis of text. In particular, the "spell" system of the game has interesting and unique characteristics as it is primarily made up of high level and descriptive text but has many of the game's main rules embedded with that text. Thus, we examine the capabilities of several models on the task of generating new content for this game, evaluating the performance through the use of both score-based methods and a survey on the best performing model to determine how the generated content conforms to the rules of the game and how well they might be used in the game.

Keywords: Generative Language Models, Game Content Generation, Dungeons & Dragons

1. Introduction

Dungeons and Dragons (D&D) [1] is a tabletop role-playing game (RPG) that boasts a large player base across the world and features a wide and detailed set of rules. It is also a game which encourages creativity and offers its players a great deal of freedom in play. In this work we are using the Fifth Edition version of the game, as it's currently the newest and most popular version of the game.

1.1. The Game

The gameplay of D&D revolves around the interaction between two groups to simulate a story taking place in a fictional world, where one group attempts to perform actions in this world and the other determines what happens when those actions take place. This back-and-forth gameplay is supported by a set of rules that gives participants a guideline how to resolve the actions and events that take place in the game. The two groups consist of a player known as the dungeon master (DM), and the rest of the participants who are referred to as players. The DM holds a position that is equal parts referee and storyteller. They are responsible for describing and simulating the world of the game for the players. Since the game has a freeform nature to it, the DM also has the responsibility of determining how to enforce or use the rules when players attempt to do something outside of the usual purview of the rules. On the other side, the players hold a relatively simple position of taking the role of characters within the world of the game and, as a player's respective character, interacting with the world that the DM describes for them. Since the DM is responsible for what occurs in the game, they also determine what, if any, extra non-official content to allow players to use. As such, a DM will tend to only allow non-official content into their game if they deem it to be fair for use. In this context, "fair" generally means that it is not so powerful or useful that it makes the game too easy or invalidates other content. In addition, for players whose primary job is to act as their character, some don't feel that the officially provided content offers enough options and will often seek content created by a third party so that the abilities of their character might better match what they envision.

1.2. Spells

One facet of D&D is its spell system, which allows players to use magical abilities during gameplay to enact some action on a being or object in the game. Every spell has eight parts, with many of them being highly dependent on others. Table 1 shows an example of spells in D&D. As seen in the table, these parts are the *title*, *level*, *components*, *casting time*, *range*, *duration*, *school*, and *description*. For this work we focus on the *title*, *range*, *duration*, and *description* as they carry the most importance to how a spell functions during gameplay. The descriptions of spells have a high variance in content, function, and length. They also have the most impact on how a spell works and what it does. Some spells have very simple descriptions that describe a single function the spell performs; others have long descriptions detailing a more complex spell that could be used in many ways. Descriptions also feature narrative language of what occurs when a character in the game casts the spell. No matter what a spell is though, every description has information about broader rules of the game embedded within it through the use of keywords and phrases.

1.3. Motivation

Due to the distinctive qualities of these pieces of text which hold both narrative and rules-oriented language

[1] https://dnd.wizards.com/what-is-dnd

Title	Phantasmal Killer
Level	4th
Components	Verbal, Somatic
Casting Time	1 Action
Range	120 ft
Duration	Concentration, 1 Minute
School	Illusion
Description	You tap into the nightmares of a creature you can see within range and create an illusory manifestation of its deepest fears, visible only to that creature. The target must make a Wisdom saving throw. On a failed save, the target becomes frightened for the duration. At the end of each of the target's turns before the spell ends, the target must succeed on a Wisdom saving throw or take 4d10 psychic damage. On a successful save, the spell ends. At Higher Levels. When you cast this spell using a spell slot of 5th level or higher, the damage increases by 1d10 for each slot level above 4th

Table 1: An example of official spells

in them, we see this as a unique type of text for the task of natural language generation. Showing that modern language models have the capability to infer rules from text and generate new content while retaining a narrative style of language is an interesting and unique task that could have broader impacts in the field. In addition, the use of natural language generation in the field of tabletop role-playing games could be a powerful tool. Since these games are based primarily around the use of written and spoken language, they could be enhanced through the use of computer generated content used before or during play to generate content for specific scenarios as the need arises. With this work we also hope to bring to light some of the unique facets of TRPGs as a medium that hold potential for future work in content generation and player interaction. (Liapis et al., 2014) presents the many interesting features of video games when it comes to the topic of computational creativity. We believe TRPGs to hold many of the same features as video games in this field, as well as other features unique to this medium.

2. Related Work

There is plenty of work in automatic text generation for video game content and character dialogue (Côté et al., 2018; Urbanek et al., 2019; Sirota et al., 2019; Ammanabrolu et al., 2020; Liu et al., 2021; Yao et al., 2020; Walton, 2020). One of the most related works to ours is (Woolf, 2019) on generating cards for the game *Magic the Gathering*[2] This work utilized a generative language model for generating cards and is like our work in that the cards contain descriptions similar to the descriptions of D&D spells but without the descriptive language. In their work they fine-tuned GPT-2 (Radford et al., 2019) to generate certain parts of a card by inserting unique marks on the parts of interest, such as using brackets around one component or parentheses around another. We adapt a similar technique to tag each part of a spell's data with the hope that the models would learn to differentiate each component and how they relate. It also serves as a convenient technique for checking which parts of a spell have been generated by examining the tags.

Besides this work, there are other works that are not quite as closely related, but still highly relevant. For instance, *AI Dungeon 2*, which utilizes the GPT-2 1.5B parameter model to generate "choose your own story" style adventure games. This model has been deployed online[3] with much success and shows the possibilities of using these sorts of models as a core gameplay mechanic.

(Ammanabrolu et al., 2019) presents methods for using a Markov Chain model and a Neural Language Model to generate game content. They use models to generate content in the form of quests that provide a player with a set of objectives to complete towards some goal. These quests center around players being given a list of ingredients and a recipe they must complete. The neural model in this work utilizes an LSTM (Hochreiter and Schmidhuber, 1997) model to generate a list of ingredients, which is then fed to GPT-2 to generate a title and a set of instructions to complete based on the ingredients. This work incorporates a human-participant study to determine a number of qualities of these quests, such as their coherence and creativity, as well as if the player felt accomplished upon completing them. This work is highly related to our own as both attempt to generate textual game content that needs to be coherent and creative through the use of a neural language model.

(Fan et al., 2020) presents a work on using machine learning (ML) algorithms to compose worlds for a text based game. In this work, worlds are described as being made up of various locations connecting with one another, with each location containing characters and objects that a player can interact with and use. The authors use multitask learning to train several models to connect pre-made locations together to make the world, populate the world with characters, and populate the world with objects. In addition to using ML to construct and populate worlds, they employ a transformer based model to generate new characters and objects which can then be placed into the newly generated

[2] https://magic.wizards.com/

[3] https://play.aidungeon.io/

<namestart> Acid Splash<nameend>
<rangestart>60 Feet<rangeend>
<durationstart>Instantaneous<durationend>
<descriptionstart>You hurl a bubble of acid. Choose one creature within range, or choose two creatures within range that are within 5 feet of each other. A target must succeed on a Dexterity saving throw or take 1d6 acid damage.

At Higher Levels. This spell's damage increases by 1d6 when you reach 5th level (2d6), 11th level (3d6), and 17th level (4d6).<descriptionend>

Table 2: An example of the training dataset

world.

3. Approach

Our approach to this work involves the following four components: 1) Create a viable dataset; 2) Train several generative models on the data; 3) Compare each model's performance using scoring metrics BLEU and BERTScore and subjective analysis of the quality of text generated; 4) Lastly, take spells from the best performing model and incorporate them, alongside human-made spells, in a survey given to players of D&D to determine player desirability and consistency with the rules. The models we decided to use for generation are an N-Gram model using Markov Chains, a model utilizing LSTM layers, and GPT-2 by OpenAI fine-tuned on our dataset.

4. Dataset

We used a collection of all 554 official D&D spells[4], as well as 2,598 player-made spells from the website http://dandwiki.com. For the player-made spells we had to create a web scraper to gather the relevant data. We then combined both datasets into a single file and removed all data irrelevant to the work such as a spell's school or components. The text for each spell then contained its *title*, *range*, *duration*, and *description*. We tagged each spell's attributes using tags such as "<namestart>" and "<nameend>" and concatenated all the attributes of a spell together into a string. We then removed 50 randomly picked spells from the entire dataset to be used later for evaluation. This left us with 3012 spells in our training set. Table 2 shows an example of our training dataset.

5. Experiments

5.1. N-Gram

Our N-gram model is a simple word-based 6-gram model that was trained with no smoothing. The training data for this model was unique from the others. The input data was not tagged but instead we placed "signifiers" such as "*title:* " and "*description:* " before each spell attribute. We then concatenated the spell attributes into a single string for each spell and used this as the training data for our model. In addition, each string was padded with tags for the beginning and end of the spell. During generation we truncate everything generated after the end tag if it has been generated.

5.2. LSTM

Based on common practice, the LSTM model we used is a standard character-based model that contains an embedding layer with dimension of 256, two LSTM layers of size 1024 and 512, respectively, dropout layers after each LSTM layer with a dropout-rate of 0.1, and lastly a dense layer the size of the vocabulary. Our vocabulary for this model contains 48 characters in total (all alphanumeric characters, in addition to 12 punctuations and symbols including !, ?, ;, :. ', ", (,), -, +, <, and >). This model was implemented using the Keras library[5]. We trained it on 200-character long sequences from our dataset for over 100 epochs and with a learning rate of 0.001. The data for this model was slightly altered, as each spell also contained tags to indicate the start and end of the spell. During generation we truncated everything generated after the end tag if it has been generated.

5.3. GPT-2

Our GPT-2 model was implemented with HuggingFace's transformers library[6]. We fine-tuned the pre-trained model offered by the library on a text file that contained every spell in our training dataset. This fine-tuning took place over 3 epochs. Our training data for this model was slightly different as we combined the tagged text for every spell into a single text file which was then given to the model. We did not use tags to signify the start and end of a spell for this model. However, during generation, we did truncate everything generated after the first tag signifying the end of a spell's description was generated.

6. Evaluation

Our evaluation contained three parts, a baseline examination using BLEU scores, an examination with BERTScore, and a survey with our best model, the GPT-2 based model.

6.1. BLEU Score

BLEU scores (Papineni et al., 2002) are generated based on the average number of overlapping one, two, three, and four-grams between a "reference" sentence and a "hypothesis" sentence. For our evaluation, we generate a hypothesis sentence by taking the first 40 tokens of a reference sentence and using that to generate the hypothesis. Since the LSTM based model uses

[4] https://www.kaggle.com/code/josephstreifel/dnd-spells/data

[5] https://keras.io

[6] https://huggingface.co

characters rather than word tokens, we use the first 200 characters of a reference sentence to generate the hypothesis. Each hypothesis is then compared to its reference sentence and all of the scores and average to find the final score of the model. We chose this as a baseline scoring metric as it's widely used in quickly measuring the quality of generated text compared to some reference text and is easy to understand and interpret.

6.2. BERTScore

BERTScore (Zhang et al., 2019) is used to measure similarity between two sentences by leveraging BERT's contextual embeddings to calculate the cosine similarity between each token in a reference sentence to each token in a predicted sentence. For this part of the evaluation we use the same reference and hypothesis sentences described above for the BLEU scores. We chose this as a scoring metric as it can more accurately capture the semantic similarity between two sentences than more naive approaches like BLEU, which is relevant for our work as there could be many ways to word a spell and have a similar result. We used the implementation provided by HuggingFace's Datasets library.

6.3. Survey

For our final evaluation we created a survey containing 5 GPT-2-made spells that we determined to be good results, and 5 randomly selected player created spells for D&D from `https://www.dandwiki.com/wiki/`. Without labeling where the spells were from and with random ordering, we asked respondents the following questions for each spell:

- "*What do you think made this?*"
- "*How well do you think this spell conforms to D&D's rules?*"
- "*Would you play/allow this spell?*"

Each of these questions were multiple choice. The first had options of "*Human*", "*AI*", and "*I've seen this spell before, and I know it was made by a human.*". The second was a choice from 1 to 5 with 1 being "*Doesn't fit in with the rules at all*", and 5 being "*would fit in right alongside official spells*". The third was also a choice from 1 to 5 with 1 being "*Definitely wouldn't*" and 5 being "*Definitely would*". The survey was posted in several D&D focused Discord servers including one for a D&D club at our institution.

7. Results

7.1. BLEU Score Results

The BLEU score results in Table 3 show that there's a much higher number of n-gram overlaps between the reference and generated spells for GPT-2 than both other models. This may indicate that GPT-2 more often uses the same words and phrases in the reference spells. Since BLEU penalizes generations that are shorter than their reference, it's worth mentioning GPT-2's tendency to generate long repetitions of text which may have pushed the results partially in its favor. However, since the other two models also occasionally had generations that were very long, we suspect GPT-2's habit of repetition did not skew the results and that the relative scores are still accurate.

Model	BLEU Score
N-Gram	6.6
LSTM	10.7
GPT-2	17.7

Table 3: BLEU score results for each model

Model	Precision	Recall	F1
N-Gram	0.756	0.852	0.801
LSTM	0.886	0.891	0.888
GPT-2	0.932	0.914	0.923

Table 4: BERTScore results for each model

7.2. BERTScore Results

The BERTScore results in Table 4 show that the semantic similarity of spells generated by GPT-2 is higher than those generated by the other models. Since the hypothesis spells were generated with only the first 40 tokens of the reference sentence, they were generated largely based off of the *title*, *range*, *duration*, and a small piece of the *description*. Due to this, the scores would indicate that the meaning and wording of the generated description is closely related to that of its reference spell's description. It's worth mentioning that unlike the other models, GPT-2's precision is higher than its recall. This indicates that many tokens generated by GPT-2 are closely related to some set of tokens in the reference spell, but not as many tokens in the reference are in that set and thus have few similar counterparts in the generated spell. This could be due to GPT-2 being pre-trained and thus having the capability to generate words that aren't in our dataset.

In addition, general observation of the spells generated by each model such as those in Table 5 demonstrates the level of ability of each model. A reading of the example from GPT-2 indicates that spells generated by GPT-2 are capable of a high level of coherence and the descriptions contain similar patterns and wording as seen in official spells. Due to the high level of performance of GPT-2 compared to the other models, we chose only spells generated by GPT-2 for use in the survey.

7.3. Survey Results

A total of 14 people responded to the survey, and all of them identified as people who play D&D. With a sample size as small as 14 responses, it's hard to glean any true conclusions from our survey. However, as you can see the results are extremely close in all categories. D&D players were successfully able to identify an AI

Model	Spell
N-Gram	Title: fire whip Duration: concentration, up to 1 hour Range: touch Description: when you cast this spell using a spell slot of 3rd level or higher, 30 seconds are added to the time the target is banished for each slot level above 4th. the creatures must be within 30 feet of each other when you target them.
LSTM	Title: Projectile Creature Range: 10 feet Duration: Instantaneous Description: A Warlice distortion of your hands in your fanging prowess, as a bonus action on a location within range you entered. Happiness points that are instantaneously runlois pit functions, along with its quantity of the caster or the caster such as metal weight.
GPT-2	Title: Magical Blade Range: 10 feet Duration: Instantaneous Description: You create a magical blade which resembles a blade of magical force. Choose a creature within range. Each creature in a 20-foot-radius sphere centered on that creature must make a Strength saving throw. On a failed save, a creature takes 5d10 force damage and is knocked prone. On a successful save, the creature takes half as much damage. At Higher Levels. When you cast this spell using a spell slot of 4th level or higher, the damage increases by 1d10 for each slot level above 3rd.

Table 5: Example spells generated by each model

	Player Made	GPT-2
Correctly Identified	57%	59%
Average Rule Conformity	3.39	3.37
Average Playability	3.47	3.49

Table 6: Survey results

generated spell slightly more than the human ones, while the AI spells were reported to conform to the D&D rules slightly less yet be slightly more playable with more respondents saying they would use/allow these spells in their games. It's also worth noting that for one AI generated spell, 2 of the 14 respondents claimed to have seen the spell before and knew for a fact it was written by a human. No other spell got this response, and it's clear the respondents hadn't seen that spell previously. We suspect that due to the simplistic nature and brevity of the spell, these respondents recognized common traits between it and other similar spells they had seen before. Given the small differences between the corresponding survey scores between human-made and GPT-2-made spells, it appears that our generated spells are of similar quality as the spells from `https://www.dandwiki.com/wiki/`.

Title	Conjure Ray of Force
Range	Self
Duration	Instantaneous
Description	You channel the power of your spirit into the ray of force and create a ray of force that is stronger than the spell's damage. The ray of force deals an extra 1d8 force damage to all targets within range. Target one creature in range. On a hit, the target takes 1d8 force damage. This additional damage is increased by 1d8 if you cast the spell at the same time every day for the past 24 hours.

Table 7: One example of problematic generation from GPT-2

8. Error Analysis

Although the spells were received relatively well by the survey participants, there are a number of problems in the spells generated by GPT-2.

The spell in Table 7 is evidence of generation that shows interesting and novel generation in the sentence, "*This additional damage is increased by 1d8 if you cast the spell at the same time every day for the past 24 hours*". This introduces a mechanic that could be interesting in play. This mechanic however is not well used in this spell, and would be better used in a spell that is unrelated to combat. In this way the model succeeds in utilizing an interesting mechanic, but fails to use it in a way that enhances the spell.

As for the spell in Table 8, its description is fine and works well on its own, however it's entirely inconsistent with the spell's range and duration. In the description it states "*You create a bolt of thunder that flies from your hand towards a willing target. Make a ranged spell attack against the target*", however the range of the spell is "*Self*". This sort of contradiction is a common problem with spells generated by this model and commonly occurs with each attribute of the spell. For example in this spell this type of problem occurs with each attribute of the spell and its description.

Title	Step into Darkness
Range	Self
Duration	1 Minute
Description	You create a bolt of thunder that flies from your hand towards a willing target. Make a ranged spell attack against the target. On a hit, the target takes 3d6 lightning damage. On a miss, the target takes half as much damage.

Table 8: Another example of problematic generation from GPT-2

9. Conclusions and Future Work

9.1. Conclusions

In this work, we explored to use 3 generative language models for automatic spell generation for the game D&D. Our results show that language models can generate texts at a comparable level to amateur designers, with descriptions that are both thematically interesting and fitting to the set of rules inferred by the text in the training corpus. This technique could be applied to other aspects of D&D or other games to generate new and interesting content. This could be used in many ways, such as an aid for designers creating new content for a game where a model might generate several suggestions to what the designer is currently writing, similar to modern email and messaging clients giving generated suggestions. Since tabletop games like D&D require extensive planning on the part of the DM, a system to generate new content for them could dramatically reduce the work required to prepare content to play the game, thus making it easier for more people to play and enjoy the game.

Generating new content in this manner could also be used during gameplay either to supplement gameplay or as a deliberate mechanic by the game's designers. This could lead to an entirely new kind of tabletop game where the content is generated dynamically as the game progresses.

9.2. Future Work

As this work is still a preliminary exploration in this field, there are several places for improvement upon both our best model and the others. The first place of improvement would be finding more data, and the second place would be pre-processing the data further. Since many of the spells used for training were sourced without regard for any sense of quality, there were likely biases and problematic sections that the models learned. Removing some of these problematic spells would be a step in the right direction for improving results. However, simply obtaining more data could resolve the issues presented by problematic spells, as the model would likely have more good spells to learn from in a larger dataset. Having a wider array of spells in the training set would also be highly beneficial, as each of the models tend to only generate spells that deal damage.

One potential method to deal with the bias towards damage-oriented spells would be to split the data apart into spells that are primarily damage focused and those that aren't, and train or fine-tune a model using each of these sets separately. This may be a good method to make up for the large differences in the two types of spells.

Using a more powerful model such as larger and more powerful versions of GPT-2 or newer architectures like GPT-3 could also yield significant gains. Since these spells can sometimes have very long descriptions that need to remember key information during the entire generation, such as the title or range, it would be useful to use a more powerful model that has a better ability to retain information.

Overall there are many places in which this task could be improved to create even better results than the already considerable ones shown here.

In summary, this work shows that modern generative language models can be a potentially powerful tool to aid the design and play of tabletop roleplaying games like D&D and any other games that rely on descriptive text that is embedded with rules.

10. Bibliographical References

Ammanabrolu, P., Broniec, W., Mueller, A., Paul, J., and Riedl, M. O. (2019). Toward automated quest generation in text-adventure games. *arXiv preprint arXiv:1909.06283*.

Ammanabrolu, P., Cheung, W., Tu, D., Broniec, W., and Riedl, M. (2020). Bringing stories alive: Generating interactive fiction worlds. In *Proceedings of the AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment*, volume 16, pages 3–9.

Côté, M.-A., Kádár, A., Yuan, X., Kybartas, B., Barnes, T., Fine, E., Moore, J., Hausknecht, M., Asri, L. E., Adada, M., et al. (2018). Textworld: A learning environment for text-based games. In *Workshop on Computer Games*, pages 41–75. Springer.

Fan, A., Urbanek, J., Ringshia, P., Dinan, E., Qian, E., Karamcheti, S., Prabhumoye, S., Kiela, D., Rocktaschel, T., Szlam, A., et al. (2020). Generating interactive worlds with text. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 1693–1700.

Hochreiter, S. and Schmidhuber, J. (1997). Long short-term memory. *Neural computation*, 9(8):1735–1780.

Liapis, A., Yannakakis, G. N., and Togelius, J. (2014). Computational game creativity. In *ICCC*.

Liu, J., Snodgrass, S., Khalifa, A., Risi, S., Yannakakis, G. N., and Togelius, J. (2021). Deep learning for procedural content generation. *Neural Computing and Applications*, 33(1):19–37.

Papineni, K., Roukos, S., Ward, T., and Zhu, W.-J. (2002). Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, pages 311–318.

Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., Sutskever, I., et al. (2019). Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.

Sirota, J., Bulitko, V., Brown, M. R., and Hernandez, S. P. (2019). Towards procedurally generated languages for non-playable characters in video games. In *2019 IEEE Conference on Games (CoG)*, pages 1–4. IEEE.

Urbanek, J., Fan, A., Karamcheti, S., Jain, S., Humeau, S., Dinan, E., Rocktäschel, T., Kiela, D., Szlam, A., and Weston, J. (2019). Learning to speak and act in a fantasy text adventure game. *arXiv preprint arXiv:1903.03094*.

Walton, N. (2020). Ai dungeon 2: creating infinitely generated text adventures with deep learning language models.

Woolf, M. (2019). How to make custom ai-generated text with gpt-2, Sep.

Yao, S., Rao, R., Hausknecht, M., and Narasimhan, K. (2020). Keep calm and explore: Language models for action generation in text-based games. *arXiv preprint arXiv:2010.02903*.

Zhang, T., Kishore, V., Wu, F., Weinberger, K. Q., and Artzi, Y. (2019). Bertscore: Evaluating text generation with bert. *arXiv preprint arXiv:1904.09675*.