AI Generated Movie Scenes

Daniel Zielinski
Jared Hagerman

October 8, 2021

1 Description

Extracting sentiment and tone from text is becoming a popular use of AI as opinions and human emotions proliferate on the Internet through social media, surveys, and online reviews. The challenge of this project is to use Natural Language Processing (NLP) techniques to take this concept a step further. We hope to train a model to detect not only sentiment but character features, such as personality, from movie scripts and project those features into an AI generated script. The primary goal is essentially to create interactions between "characters."

Consider this interaction from the movie *2001: A Space Odyssey* (Kubrick, 1968) between the human, Dave, and the ship's computer, Hal:

> **Hal:** Naturally Dave I'm not pleased that the AO-unit has failed but I hope at least this has restored your confidence in my integrity and reliability. I certainly wouldn't want to be disconnected even temporarily as I have never been disconnected in my entire service history.
> **Dave:** I'm sorry about the misunderstanding Hal.
> **Hal:** Well don't worry about it.
> **Dave:** And don't you worry about it.
> **Hal:** Is your confidence in me fully restored?
> **Dave:** Yes it is Hal.
> **Hal:** Well that's a relief. You know I have the greatest enthusiasm possible for the mission.

The relationship between Dave and Hal is a fascinating one and has secured its place in cinematic history, not only because of what is being said, but what is not being said. Through the course of the film, Dave and Hal start to slowly distrust one another and eventually are openly antagonistic. However, for most of the movie we as viewers can detect more and more that there's a dynamic of this relationship beyond the words that are being expressed. The creepiness of

Hal is one of the defining characteristic of his persona and it is with this project that we hope to capture and model that sort of uniqueness of movie personas.

1.1 Current Solutions

AI generated text in the style of a particular persona is not a new idea. A quick search of in the Internet reveals similar projects, some of which are quite entertaining, such as an algorithm that was given 1000 hours of Batman videos and generated a scene where Batman interacted with the Joker, Two-Face, Alfred, and Robin (Patti, 2019). What makes the output of this algorithm so entertaining is that the script captures features of the characters quite well and translates each of their uniqueness into the typical dynamic between each of the characters. The hope of this project is to develop and expand this effort beyond Batman and into a wider array of personas.

A project specific to our dataset, started on Kaggle by Swaroop Kallakuri (Kallakuri, 2017), had a similar goal in that it used a Keras Seq2Seq model to continue movie dialogues after receiving an initial prompt. However, this user was unable to arrive at their desired outcome and interpreted the text generated by this model as: "*That's gibberish and rubbish ;). A lot to change and modify. Seq2Seq is not sufficient to answer these kind of dialogues or we change the architecture and re-train. If it helps you learn something and like it, please upvote and motivate me to write and share more!!*"."

This user's model was not only unable to produce the desired results but was also inefficient and troublesome to run. Some of this user's data clean up code could be quite useful and allow us to focus time on improving the model itself and tweaking the output to hopefully produce a similar effect to that of Keaton Patti.

2 Dataset

The primary dataset we intend to use was originally published for a project at Cornell University (Danescu-Niculescu-Mizil & Lee, 2011) and was later made available on Kaggle for public use(Cornell, 2017). The data is described as containing the following:

- Conversations 220,579 conversational exchanges between 10,292 pairs of movie characters
- Characters 9,035 characters from 617 movies
- Utterances 304,713 utterances

- **Movie Metadata** Genres, Release Years, IMDB Ratings, Number of IMDB Ratings
- **Character Metadata** Gender, and Position in Credits

The dataset can be technically described as about 30Mb in total size across five files, each containing either the raw text of the movie script, or meta information of how a line of text fits into the context of other lines and characters. The titles of each of these files is as follows:

- movie_characters_metadata.tsv
- movie_conversations.tsv
- movie_lines.tsv
- movie_title_metadata.tsv
- raw_script_urls.tsv

A description of how the data was collected has also been provided (Cornell, 2017):

> We started from raw publicly available movie scripts (sources acknowledged in raw_script_urls.txt). In order to collect the metadata necessary for this study and to distinguish between two script versions of the same movie, we automatically matched each script with an entry in movie database provided by IMDB (The Internet Movie Database; data interfaces available at http://www.imdb.com/interfaces). Some amount of manual correction was also involved. When more than one movie with the same title was found in IMBD, the match was made with the most popular title (the one that received most IMDB votes)
>
> After discarding all movies that could not be matched or that had less than 5 IMDB votes, we were left with 617 unique titles with metadata including genre, release year, IMDB rating and no. of IMDB votes and cast distribution. We then identified the pairs of characters that interact and separated their conversations automatically using simple data processing heuristics. After discarding all pairs that exchanged less than 5 conversational exchanges there were 10,292 left, exchanging 220,579 conversational exchanges (304,713 utterances). After automatically matching the names of the 9,035 involved characters to the list of cast distribution, we used the gender of each interpreting actor to infer the fictional gender of a subset of 3,321 movie characters (we raised the number of gendered 3,774 characters through manual annotation). Similarly, we collected the end credit position of a subset of 3,321 characters as a proxy for their status.

3 Deep Learning process

3.1 Architectures of Consideration

Recursive Neural Network(RNN) - A type of feed-forward neural network with a memory that is particularly useful for sequential data, such as audio or text (Choubey, 2020). Generic RNNs can be constructed a few different ways depending on the number of inputs and desired output(s): one-to-one, one-to-many, many-to-one, or many-to-many. Many:many is the generic structure pertinent to this project.

Sequence to Sequence (Seq2Seq) - A family of RNN architectures which includes GRUs and LSTMs (Sutskever et al., 2014).

Gated Recurrent Unit (GRU) - A sub-type of an RNN that has operations inside to act as two types of gates: 1)reset gate, and 2)update gate (Pedamallu, 2020).

Long Short-Term Memory (LSTM) - A similar type of architecture to GRUs except there are two additional types of gates: 3)forget gate, and 4) output gate (Pedamallu, 2020).

It is almost certain that we will use a Seq2Seq type of architecture, either as a GRU or LSTM. We'll investigate how a vanilla many-to-many RNN performs but we believe it's unlikely it will have sufficient memory to sufficiently label the complexities of the texts we're focusing on. Our final decision will depend on the performance we're able to get from each of these architectures.

3.2 Other Outside Code of Consideration

Transferred Learning - Transferred learning is the concept of applying knowledge learned in one context to a different but related context (Pratt, 1992). Transferred learning will offer an important component of our desired output because if we are able to capture unique characteristics of personas, then the use of Transferred Learning will allow us to manifest how those personas behave during interactions with novel personas. For example, The Joker, from Batman movies, would never interact with Hal, from 2001: A Space Odyssey, but with the concept of Transferred learning, this hypothetical interaction could be generated. Transferred learning can be implemented within a model using a Discriminability-Based Transfer (DBT) algorithm.
Transformers/BERT - Bidirectional Encoder Representations from Transformers (BERT) is a publicly available pre-trained Compact Model (Devlin et al., 2018) developed as a Google project with the goal of enabling institutions with fewer computational resources to improve modeling capabilities. BERT has been trained using the Masked Language Modeling (MLM) and Next Sentence Prediction (NSP) objectives. It has been identified as an important develop-

ment in NLP because it allows for smaller architectures to be competitive with more elaborate methods (Turc et al., 2019).

The authors of BERT have stated: "*The pre-trained BERT model can be fine-tuned with just one additional output layer to create state-of-the-art models for a wide range of tasks, such as question answering and language inference, without substantial task-specific architecture modifications*" (Devlin et al., 2018). It is because of this we expect BERT to be useful to this project as our focus is on interactions between characters. The BERT model offers a simple method for the "characters" to answers questions, make inferences, and effectively interact with other "characters." An MLM approach, as used in BERT, seems more pertinent to our goal than that of a Casual Language Model (CLM). While an MLM is bidirectional, a CLM is unidirectional meaning that MLM is preferred when trying to represent inputs whereas a CLM is preferred when generating fluent text.(Mishra, 2019)

4 Training/Testing Strategy

Sequence to sequence (Seq2Seq) will likely be the method chosen for this project due to wanting to have our model have sentences (or sequences of words) as the output. The way these models work is by taking a sequence as an input (words, audio, video) and decoding the context of the input to encode the output sequence (Dugar, 2021). We will build a model to train on a dialogue between two characters to learn how they interact with each other and the context of their responses back and forth. From there, we will build a testing model where we will give either a sentence or a few words and see what the model outputs as a response.

4.1 Evaluation Metrics

One of the most popular and accurate ways to evaluate a NLP/RNN model is with the Bilingual Evaluation Understudy score (BLEU) (Brownlee, 2019). With this evaluation method, there is a sub-function called sentence_bleu that gives a score for each sentence generated when compared to the actual sentence. With this in mind, we will be able to take the results from our testing model and compare them directly to what the output should have been and get a grade on how accurate each sentence is. If the words match perfectly, we will achieve a score of 1.0. If there is even one word off, the score will drop automatically.

From using this scoring technique, we will be able to determine if our model has been trained on enough data to generate accurate dialogue with the character of choice or if there is more data needed. We can run the training several times as well to try and improve these scores. If we wanted to generate an entire paragraph instead of just a sentence, we could also use the corpus_bleu to grade that against what the output should have been.

4.2 Expected Results

As we build and run these models, our goal is to generate dialogue between two characters that makes sense. From the existing dataset, we know that other users were only able to generate gibberish with his model and needed some tweaking and cleaning of his model to improve on the performance. We don't expect the output to match perfectly with what the original script would be but the context behind the conversation and the style of wording chosen should match closely with how the characters would normally speak to one another.

References

Brownlee, J. (2019). *A gentle introduction to calculating the bleu score for text in python*. Retrieved September 26, 2021, from https://machinelearningmastery.com/calculate-bleu-score-for-text-python/

Choubey, V. (2020). *Understanding recurrent neural network (rnn) and long short term memory (lstm)*. Retrieved September 26, 2021, from https://medium.com/analytics-vidhya/undestanding-recurrent-neural-network-rnn-and-long-short-term-memory-lstm-30bc1221e80d

Cornell. (2017). *Movie dialog corpus*. Retrieved September 26, 2021, from https://www.kaggle.com/Cornell-University/movie-dialog-corpus

Danescu-Niculescu-Mizil, C., & Lee, L. (2011). Chameleons in imagined conversations: A new approach to understanding coordination of linguistic style in dialogs. *Proceedings of the Workshop on Cognitive Modeling and Computational Linguistics, ACL 2011*.

Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2018). Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Dugar, P. (2021). *Attention - seq2seq models*. Retrieved September 26, 2021, from https://towardsdatascience.com/day-1-2-attention-seq2seq-models-65df3f49e263

Kallakuri, S. (2017). *Movie dialog generation - keras model*. Retrieved September 26, 2021, from https://www.kaggle.com/ksjpswaroop/movie-dialog-generation-keras-model

Kubrick, S. (1968). 2001: A space odyssey. *United States : Metro-Goldwyn-Mayer*.

Mishra, P. (2019). *Understanding masked language models (mlm) and casual language models (clm) in nlp*. Retrieved September 26, 2021, from https://towardsdatascience.com/understanding-masked-language-models-mlm-and-causal-language-models-clm-in-nlp-194c15f56a5

Patti, K. (2019). *I forced a bot to watch over 1,000 hours of batman...* Retrieved September 26, 2021, from https://twitter.com/KeatonPatti/status/1161284670601990146

Pedamallu, H. (2020). *Rnn vs gru vs lstm*. Retrieved September 26, 2021, from https://medium.com/analytics-vidhya/rnn-vs-gru-vs-lstm-863b0b7b1573

Pratt, L. (1992). Discriminability-based transfer between neural networks. *NIPS Conference Proceedings*. https://proceedings.neurips.cc/paper/1992/file/67e103b0761e60683e83c559be18d40c-Paper.pdf

Sutskever, I., Vinyals, O., & Le, Q. V. (2014). Sequence to sequence learning with neural networks. *CoRR*, *abs/1409.3215*. http://arxiv.org/abs/1409.3215

Turc, I., Chang, M., Lee, K., & Toutanova, K. (2019). Well-read students learn better: The impact of student initialization on knowledge distillation. *CoRR*, *abs/1908.08962*. Retrieved September 26, 2021, from http://arxiv.org/abs/1908.08962