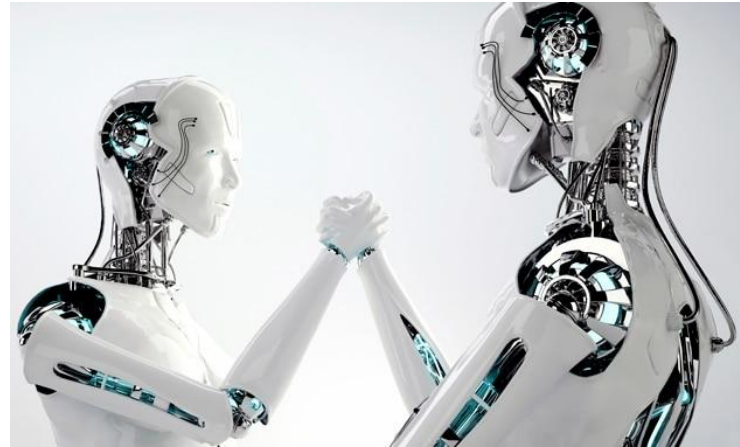


Evaluating the Logical Coherence of Conversations between GPTs

Bay Hodge



Summary of Project

Step 1: Fine-tune a pre-trained conversational language model from Hugging Face on a desired dataset.

Step 2: Then, do the same, but on a second dataset that **differs fundamentally in content, quality, and/or length**.

Step 3: Generate a conversation between these two models by feeding one's output as the other's input, and so on.

Step 4: Evaluate the logical coherence of the resulting conversation.

Step 5: Repeat, using different datasets.

Progress

- Base model: [DialoGPT](#), “a SOTA large-scale pretrained dialogue response generation model for multiturn conversations.”
- Fine-tuned DialoGPT to create:
 - RandomGPT: trained on 10,000 sentences generated by randomly sampling words out of a [dataset containing 370,105 English words](#).
 - StarWarsGPT: trained on [every line of dialogue](#) from *Star Wars: Episodes IV-VI*.
 - ShakespeareGPT: trained on “[Tiny Shakespeare](#),” 40,000 lines of dialogue from a variety of Shakespeare plays.
- Use GloVe embeddings to measure the average similarity score of a conversation.

Example: StarWarsGPT with StarWarsGPT

```
>> Please input a prompt to get the conversation going: come on, we have to run!  
StarWarsGPT-small (Model #1): I'm on it, boss!  
StarWarsGPT-small (Model #2): We're on it!  
StarWarsGPT-small (Model #1): That's it! Chewie, pull up!  
StarWarsGPT-small (Model #2): Oh, no!  
StarWarsGPT-small (Model #1): Get up! We're coming in much faster this time.  
StarWarsGPT-small (Model #2): We've got to get out of here quick.  
StarWarsGPT-small (Model #1): I've got a bad feeling about this.  
StarWarsGPT-small (Model #2): !!!  
StarWarsGPT-small (Model #1): !  
StarWarsGPT-small (Model #2): Good shot, kid.
```

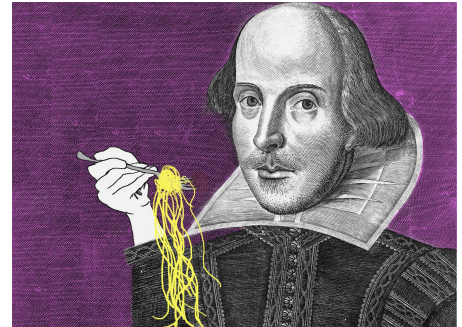
Average similarity score: 0.8017039924860001



Example: ShakespeareGPT with ShakespeareGPT

```
>> Please input a prompt to get the conversation going: come on, we have to run!  
ShakespeareGPT-small (Model #1): Go, go, go with us; we must go with you.  
ShakespeareGPT-small (Model #2): Go with me, and let me see them.  
ShakespeareGPT-small (Model #1): I will not.  
ShakespeareGPT-small (Model #2): Come, go.  
ShakespeareGPT-small (Model #1): Do not look to the sky.  
ShakespeareGPT-small (Model #2): It is the wind that blows so wildly.  
ShakespeareGPT-small (Model #1): !!!?!!  
ShakespeareGPT-small (Model #2): !???  
ShakespeareGPT-small (Model #1): !,!!!,?!,,!,.!!!.,'!!!'!?,!?.!?'!,!'!!!!'!?!',  
ShakespeareGPT-small (Model #2): !.,!.
```

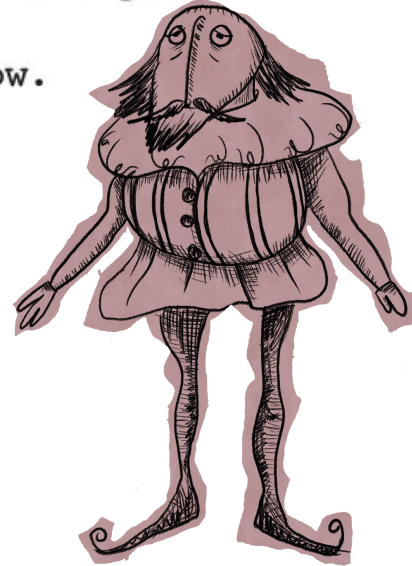
Average similarity score: 0.8953382670879364



Example: StarWarsGPT with ShakespeareGPT

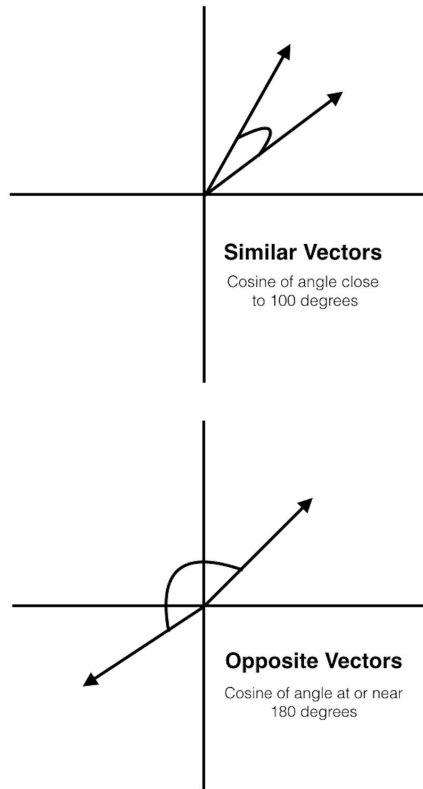
```
>> Please input a prompt to get the conversation going: come on, we have to run!  
StarWarsGPT-small (Model #1): Yes, sir.  
ShakespeareGPT-small (Model #2): Come on, I say.  
StarWarsGPT-small (Model #1): Oh, my.  
ShakespeareGPT-small (Model #2): 'Tis no time to jest now, I warrant you.  
StarWarsGPT-small (Model #1): What's so funny?  
ShakespeareGPT-small (Model #2): The humour is gone, sir, for now.  
StarWarsGPT-small (Model #1): !!!!?!!  
ShakespeareGPT-small (Model #2): !?  
StarWarsGPT-small (Model #1): !  
ShakespeareGPT-small (Model #2): ?!??!
```

Average similarity score: 0.7754686375459036



Evaluation by Averaging Similarity Scores of Sentences

```
def average_similarity_score(sentence_list):  
    # Compute similarity score between each pair of adjacent sentences in the list and return their average  
    similarity_scores = []  
  
    # Iterate through adjacent sentence pairs  
    for i in range(len(sentence_list) - 1):  
        sentence1 = sentence_list[i]  
        sentence2 = sentence_list[i + 1]  
  
        # Preprocess and tokenize sentences  
        tokens1 = word_tokenize(sentence1.lower())  
        tokens2 = word_tokenize(sentence2.lower())  
  
        # Get glove word embeddings for each token  
        sentence1_embeddings = [glove[token] for token in tokens1 if token in glove.stoi]  
        sentence2_embeddings = [glove[token] for token in tokens2 if token in glove.stoi]  
  
        # Compute sentence embeddings by averaging word embeddings  
        sentence1_embedding = torch.mean(torch.stack(sentence1_embeddings), dim=0)  
        sentence2_embedding = torch.mean(torch.stack(sentence2_embeddings), dim=0)  
  
        # Compute cosine similarity  
        similarity = F.cosine_similarity(sentence1_embedding.unsqueeze(0), sentence2_embedding.unsqueeze(0))  
  
        similarity_scores.append(similarity.item())  
  
    # Calculate average similarity score  
    average_similarity = sum(similarity_scores) / len(similarity_scores)  
  
    return average_similarity
```



Next Steps

- Train additional models and evaluate conversations between them
- Improve existing models by tweaking hyperparameters
- Experiment with [Coh-Metrix](#), “which analyzes texts on over 200 measures of cohesion, language, and readability” as a better way of evaluating logical coherence
- In preparation for the final paper, achieve rigorous understanding of the mechanics behind pretrained, conversational models

Sources

Model training and fine-tuning:

- <https://huggingface.co/microsoft/DialoGPT-small>
- <https://www.youtube.com/watch?v=eereDVZeE7Y>
- <https://github.com/MexJason/HubermanGPT>
- <https://nathancooper.io/i-am-a-nerd/chatbot/deep-learning/gpt2/2020/05/12/chatbot-part-1.html>

Data:

- <https://www.kaggle.com/datasets/xvivancos/star-wars-movie-scripts>
- <https://raw.githubusercontent.com/karpathy/char-rnn/master/data/tinyshakespeare/input.txt>
- <https://github.com/dwyl/english-words/tree/master>