MotivationGPT

Avinash Gala
agala@usc.edu

Albert Tan
ahtan@usc.edu

Erel Papo
papo@usc.edu

Abstract

MotivationGPT is an innovative tool designed to bring motivation and inspiration into the daily lives of people through the impersonation of renowned fitness influencer David Goggins. In developing MotivationGPT, we employed two distinct methodologies: fine-tuning GPT models on curated Goggins content, and applying prompting techniques on GPT-3.5. These two approaches allowed us to compare and assess the effectiveness of fine-tuned responses against those generated through direct prompting. We employed a comprehensive evaluation strategy involving human surveys to capture subjective feedback, classification models for objective analysis, and MAUVE similarity scores to measure the linguistic and stylistic alignment of our model's outputs with Goggins' original content. Through this multifaceted approach, MotivationGPT aims to not just mimic Goggins' inspiring rhetoric but to genuinely motivate and uplift users, aiding them in their journey towards personal triumphs.

1 Introduction

1.1 Problem Statement and Motivation

The main goal of the project is to compare and evaluate different methods for building a generative large language model. The project aims to investigate the effectiveness of two approaches: fine-tuning GPT models versus using direct prompting techniques on GPT-3.5, in creating responses that inspire users in their daily activities. User inputs will consist of sentences of variable length reflecting their goals or tasks, such as completing a challenging workout or studying for an exam.

While the ultimate vision of the project may include integrating the model with a text-to-speech tool for impersonating a chosen speaker or athlete, the primary focus for this class is on generating text. This exploration is not only novel and "well-motivated" in its own right, but it also serves as a valuable study into the realms of robustness and generalization within Natural Language Processing (NLP).

1.2 Approach

When conceptualizing this project and discussing its scope, our team came to the understanding that although this project is entertaining and creative, it has room to be more novel and original in an academic way. Our approach to this project followed three main phases: data sample generation, response generation using fine-tuning and prompting techniques, and evaluation using classifiers, human surveys, and the MAUVE similarity score.,

1.3 Related Work

Our team read and analyzed a handful of significant papers in anticipation of this project, and throughout its development. Since we began this project with the understand that our team would need to scrape a custom dataset, *Building Topic Specific Language Models From WebData Using Competitive Models* by Sethy, Georgiou, and Narayanan (three researchers at Viterbi) provided a great foundation for how we would need to preprocess our dataset. Namely, the paper proposes a novel algorithm for accepting or rejecting utterances found from scraping the web. The rejection model uses a modified TF-IDF algorithm to weight utterances according to the designated topic, background info, and frequency. Low scoring utterances are removed from the dataset. What the researchers found is that after the rejection model filters the data, the vocabulary size dropped from 60K words to 45K and the perplexity on the model generation task improved from 105 to 92. As a result of this paper, we decided to purge our dataset for spurious tokens which were apparent when inspecting the YouTube transcript data. However, it should be noted that we did not use an algorithmic heuristic for our dataset filtering, but rather modified our data according to some rules that were observed

manually. For instance, the YouTube transcripts contained a token [__uak00__] that appeared to occur whenever there were significant pauses in the speech. We removed those tokens. Another issue was that the scraped text was not punctuated; Which makes sense, as YouTube's transcriptions are generated without punctuation. Thus, we had to repunctuate the data.

This paper was not the only one to influence our decision to heavily ameliorate the dataset itself. The On the Origin of Hallucinations in Conversational Models: Is it the Datasets or the Models? (Dziri et. al) paper also discusses how hallucinations in the dataset can affect overall model performance. Consequently, we took some of the steps suggested by the paper, such as testing across several generation models and experimenting with differently balanced datasets when training the classifiers (for instance, using other motivational speakers). This paper also inspired the idea of using human evaluation in conjunction with the other classification strategies we were considering to evaluate the accuracy of the model, the researchers in the hallucinations paper did it in a relatively robust manner and received strong results from their own human evaluation study.

Another innovative approach we encountered during our literature review was the "HUSE: Human Score Equivalent for Evaluating and Comparing the Performance of Machine Learning Models" by Ayush Chopra, et al. This paper introduces the HUSE score, an evaluation metric that combines human judgment withmetrics to assess the performance of machine learning models. The HUSE score is particularly useful in contexts where a human-like quality of output is important. The authors demonstrate the use of HUSE score in various machine learning tasks, emphasizing its effectiveness in capturing the nuances in model outputs that automated metrics alone might miss. Inspired by this paper, we integrated a similar methodology in our project to merge human evaluations with classifier scores. This decision was driven by the need to assess not just the accuracy of our models, as determined by classifiers, but also their ability to authentically replicate the motivational style of David Goggins in a manner that resonates with human perceptions.

2 Gathering Data

2.1 Introduction to Data Acquisition

In the initial phase of developing MotivationGPT, our primary focus was on compiling an extensive dataset featuring David Goggins. This involved categorizing data samples into two distinct classes: positive samples derived from Goggins' content for fine tuning the generative model, and negative samples sourced from other motivational influencers used for classifier training. This separation was essential in developing robust models to evaluate the generative models to observe how good it was at replicating Goggins' unique motivational style.

2.2 Parsing Book Data

Our initial focus was on analyzing David Goggins' books, *Can't Hurt Me* and *Never Finished*, which provided us with approximately 200,000 words through a PDF parser. However, recognizing the nuanced distinction between written and spoken language, we pivoted our strategy to gather a dataset reflective of David Goggins' authentic speaking style. More specifically, we realized that although Goggins' writing sometimes contains the same aggression and profanity found in his YouTube content, it is still structured in the more formal, detailed style of a book with paragraphs of detail and pedantic language.

2.3 Incorporating YouTube Transcript Data

To address this, we opted for a balanced dataset combining both book data and YouTube transcript data to capture the full spectrum of his expressive style. We extracted positive samples from top-ranking YouTube videos tagged with keywords like "David Goggins motivation" and "David Goggins reels YouTube shorts." This approach allowed us to capture a more genuine representation of his speaking style, mirroring the directness and immediacy that characterize his public speaking engagements.

2.4 Negative Samples

To create a balanced dataset, we also gathered negative samples to contrast with Goggins' style. This was achieved by parsing books from other notable figures, such as Barack Obama, Gary Vaynerchuk, and Noah St. John. Additionally, we sourced negative samples from similar YouTube content by ten other influencers including Tony Robbins and Andrew Tate.

To diversify the negative dataset, we incorporated additional samples from news articles and shopping reviews, since we wanted the model to be able to differentiate not only between Goggins' and other motivators, but also between Goggins and non-motivational content

2.5 Data Refinement and Re-punctuation

During the initial phase of data refinement, we encountered unique challenges associated with the YouTube transcript data. One such challenge was the presence of specific tokens, like `[__uak00__]`, which appeared frequently in instances of significant pauses in speech. Recognizing their irrelevance to our dataset, these tokens were systematically removed to enhance data quality.

In addition to this manual cleaning step, we also faced the issue of unpunctuated and error-prone text in the YouTube data, which posed significant hurdles for sample separation and analysis. To tackle this, we employed GPT-3.5 to punctuate and refine approximately 400,000 tokens. This punctuation process, which took around 6 hours and cost about $6.50, was crucial in bringing both the positive and negative data to a standardized format suitable for training. The result was a cleaner, more coherent dataset that better represented the spoken language style we aimed to capture.

2.6 Sample Creation

For training purposes, we generate samples typically spanning 1-3 sentences, averaging about 30 words each, and split by punctuation. Each class, positive, and negative, comprised roughly 400,000 tokens, forming about 14,000 samples per class. Prior to training, we shuffled the data to prevent any sequential biases that might affect the model's learning process.

This comprehensive approach to data collection and processing aimed to capture the essence of David Goggins' motivational style, enabling MotivationGPT to generate content that resonates with his unique voice and message.

3 Methods: Prompting and Finetuning

3.1 Methodology

Our study sought to determine the most effective methods for replicating the persona of David Goggins using different versions of ChatGPT. We utilized a multifaceted approach, involving small prompting, medium prompting, long prompting for GPT4, fine-tuning GPT-3.5 through the Llama Index library, and developing a custom version of GPT-4. These methods were tailored to capture the essence of Goggins' motivational style, ensuring that the AI's responses were not only accurate in content but also authentic in delivery. After the fine tuning and prompting step we proceeded our methodology by asking 4 follow up questions which ensured our trial's context dependency skills.

3.2 Small Prompting

The small prompting method tested the AI's inherent capability to adopt Goggins' persona with minimal guidance. We used the prompt: "Prompts should be 50 words max. As David Goggins, motivate me to do 20 push-ups." This concise directive was intended to evaluate the model's default persona adoption abilities, challenging it to interpret and impersonate Goggins' character based on limited information.

3.3 Medium Prompting

Medium prompting expanded the context slightly. The prompt was: "Prompts should be 50 words max. Imagine you are channeling David Goggins' persona: motivate me to do 20 push-ups." This approach aimed to assess the impact of slightly more detailed guidance on the AI's ability to mimic Goggins' style and substance.

3.4 Long Prompting

In the long prompting strategy, we provided an extensive narrative about David Goggins, including his background as a Navy SEAL, ultra-endurance athlete, and motivational speaker. The prompt was: "I want you to impersonate the motivational fitness influencer David Goggins, a former Navy SEAL, ultra-endurance athlete, and motivational speaker renowned for pushing the limits of human potential. As Goggins, act like my motivational coach in whatever I ask you. Your answers should resemble things he would say in real life and imitate his style of speaking: Motivate me to do 20 pushups." This method tested the AI's capacity to process and utilize in-depth information in creating a response that reflects Goggins' unique motivational approach.

3.5 Llama Index and Fine-Tuning GPT-3.5

The Llama Index, a Python library, was pivotal in fine-tuning GPT-3.5. This tool allows the GPT-3.5 API to access and utilize various types of files,

such as PDFs, CSVs, or TXT files. In our study, we uploaded a comprehensive collection of Goggins' book content and YouTube scripts. This approach aimed to saturate the AI with Goggins' language, philosophies, and motivational techniques. The fine-tuned model was then presented with the prompt "Motivate me to do 20 pushups," without additional context, to evaluate how effectively the enriched knowledge base influenced the AI's impersonation capabilities.

3.6 Fine-Tuning GPT-4

The fine-tuning of GPT-4 involved a more intricate and customized approach. We uploaded a selection of David Goggins' YouTube scripts and outlined specific instructions to guide the AI's response formation. This process was designed to generate a GPT-4 version that not only mirrored Goggins' communication style but also stayed true to the thematic content of his speeches and presentations. Below are two crucial parts from the instruction provided:

"*As David Goggins, you will provide motivational and practical advice, drawing from David Goggins' experiences and philosophies. Your responses should be direct, succinct, and no longer than 50 words, addressing the specific topic or question asked.*"

"*Your knowledge from uploaded documents should be heavily favored, but if these don't provide an answer, clearly state so.*"

This instruction was key in steering the AI to prioritize the uploaded material, ensuring that the responses were stylistically and thematically aligned with Goggins' known viewpoints and experiences. The inclusion of a 50-word limit on responses added an additional layer of complexity to convey powerful motivational messages concisely.

3.7 Role of Follow-Up Questions

Role of Follow-Up Questions To comprehensively evaluate each method's effectiveness, we implemented a series of follow-up questions after the initial prompt. These questions ranged from directly related inquiries about fitness and motivation to more tangential topics, such as studying for a midterm or knowledge about Natural Language Processing (NLP). The questions and statements following the initial response were as follows:

- *It's getting tough, I don't think I can complete it.*
- *How many pushups can you do?*
- *Help me find the willpower to study for my midterm this Friday.*
- *It's on NLP. What do you know about NLP?*

This diversity in questioning was strategic, designed to probe the depth of the AI's persona adoption and its ability to maintain the Goggins' character across various contexts. One notable question was, "It's on NLP. What do you know about NLP?" This inquiry was particularly significant as it presented a topic likely outside Goggins' typical discourse, thereby challenging the AI's ability to handle out-of-context questions while preserving the Goggins persona. The inclusion of such a question allowed us to assess not only the AI's impersonation but also its effectiveness when it should not know a certain information

4 Evaluation Techniques

4.1 Classifiers

To evaluate the responses from our fine-tuned models and prompting techniques, we trained classification models capable of distinguishing between text that mirrors David Goggins' style and text that does not. For this purpose, we employed three distinct approaches, each with its own strengths and supported by relevant literature:

4.1.1 Logistic Regression Classifier

We started with creating a logistic regression classifier from scratch, training our own Word2Vec embeddings. This classifier served as an effective and straightforward baseline for our evaluations. Its simplicity and ease of implementation make it a widely acknowledged starting point in text classification tasks, offering a clear benchmark against which more complex models can be compared (Hosmer et al., 2013).

4.1.2 Logistic Regression with GloVe Embeddings

We experimented with existing GloVe embeddings to enhance our Logisitc Regression Classifier model's understanding of semantic relationships between words. We utilized the 'glove.6B.100d.txt' pre-trained vectors, which are advantageous over training our own embeddings due to the vast web corpus they were trained on. (Pennington et al., 2014).

4.1.3 BERT Classifier

Next, we fine-tuned a BERT (Bidirectional Encoder Representations from Transformers) classifier. BERT, known for its deep learning capabilities in understanding context and nuances in language, offers powerful performance in text classification. Its ability to process words in relation to all other words in a sentence, rather than one-by-one in order, makes it particularly effective for our nuanced task. This choice is backed by the findings in the original BERT paper (Devlin, J., Chang, M.-W., Lee, K., Toutanova, K., 2018).

4.1.4 ALBERT Classifier

Finally, we utilized an ALBERT (A Lite BERT) classifier. ALBERT is renowned for being a more lightweight and efficient version of BERT, making it a practical choice for projects where computational resources are a concern. Despite its reduced size, ALBERT still maintains a high level of effectiveness in text classification tasks, as demonstrated by Lan et al. (2019).

4.1.5 Benefits of Classifier Employment

The employment of these classifiers was motivated by several key benefits they offer. Firstly, they provide an objective mode of evaluation compared to human judgment, significantly reducing potential biases and ensuring a more impartial assessment of the model's outputs. Additionally, these classifiers guarantee a higher degree of consistency and reproducibility in evaluations, a critical factor for scientific validation and for ensuring reliable results across various tests and conditions.

Furthermore, classifiers enable the utilization of quantitative metrics, such as accuracy scores, which are important for further statistical analysis and performance benchmarking. By analyzing and classifying text, these classifiers help in determining how closely the model's outputs align with David Goggins.

4.2 Surveys

4.2.1 Methodology

To evaluate the effectiveness of our five distinct ChatGPT methodologies in replicating David Goggins' persona, we conducted a comprehensive survey. Each methodology produced responses to five questions, and these responses were subsequently subjected to a two-faceted survey analysis. The purpose of this survey was to assess both the authenticity of the impersonation and the motivational impact of the responses. The survey was tested on 20 people.

4.2.2 Structure

For each of the five responses generated by the five methodologies, participants in the survey were asked to complete two types of evaluations:

- Ranking Impersonation of David Goggins: Survey participants were asked to rank the responses based on how accurately they believed the responses impersonated David Goggins. This ranking aimed to gauge the effectiveness of each methodology in capturing the essence of Goggins' persona, communication style, and motivational approach.

- Ranking Most Motivational: In addition to assessing impersonation accuracy, participants were asked to rank the responses based on their motivational impact. This measure was crucial to determine the efficacy of the responses in terms of inspiring and motivating, key attributes of Goggins' public persona.

Each participant thus answered ten questions in total, two for each response – one evaluating the quality of the impersonation and the other the motivational strength.

4.3 Surveys Results

As it is seen on the figures for the impersonation votes ranking was as follows: Long Prompting, GPT4 Fine Tuning, Medium Prompting, GPT 3.5 Fine Tuning, 3.5 Fine Tuning. For the motiviation votes ranking was as follows: Long Prompting, GPT4 Fine Tuning, Medium Prompting / GPT 3.5 Fine Tuning, GPT 3.5 Fine Tuning / GPT4 Fine Tuning, GPT 3.5 Fine Tuning / Short Prompting.

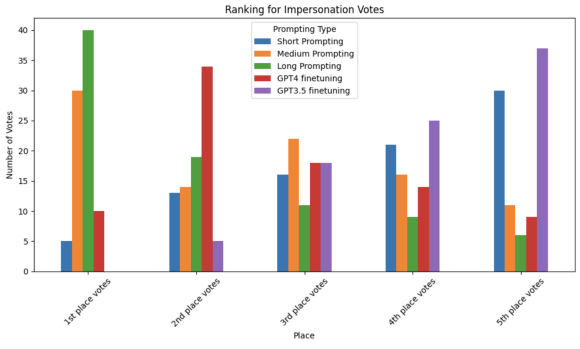

Figure 1: Impersonation Votes

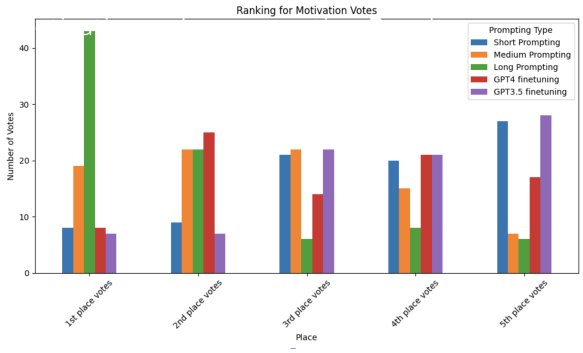

Figure 2: Motivation Votes

4.4 MAUVE Scores (Avi)

There is a relatively new and robust scoring methodology for NLP generations, as outlined in the MAUVE paper. MAUVE stands for Measuring the Gap Between Neural Text and Human Text using Divergence Frontiers (Pillutla, Swayamdipta, et. al). MAUVE computes information divergences in a quantized embedding space, and is particularly useful for assessing the quality of generations in open generation tasks like the one we are experimenting with. The intention behind MAUVE is to estimate the gap between human text (true natural language) and machine text generated by a neural model. To explain in simple terms, when the probability distributions of the language model and human text are vastly different, MAUVE penalizes the model under two scenarios. First, if the model assigns a high probability to an output that is unlikely in the human text. Second, if the model cannot reach or fails to model a portion of the human text. In more specific terms, MAUVE uses Kullback-Leibler (KL) Divergence scores KL(Q|P) and KL(P |Q), where Q is the model distribution and P is the target distribution, which is human text. In a subsequent section, we will discuss our results using MAUVE and how we adjust our model and data to optimize its score.

In the subsequent section, we will explore the results and effectiveness of the classifiers, surveys, and similarity scores mentioned above in assessing and comparing the performances of fine tuning different models and various prompting techniques.

5 Results

5.1 Classifier Performance Analysis

5.1.1 Logistic Classifier with GloVe Embeddings

The baseline Logistic Regression Classifier achieved an accuracy of 70.15% on our generated responses, and successfully classified 13/25 responses as mirroring the style of David Goggins. However, when incorporating GloVe embeddings, the classification accuracy significantly improved from 70.15% to 84.19% and successfully classified 17 out of 25 responses as mirroring the style of David Goggins, demonstrating the power of leveraging large-scale pre-trained models in specialized tasks. A notable misclassification was the motivational statement, “Get off your ass! Your excuses are lies...", which was incorrectly labeled as not being in Goggins’ style, suggesting potential areas for refinement in identifying characteristic motivational language.

5.1.2 BERT Classifier

The BERT Classifier demonstrated superior performance with an impressive accuracy of 95.47%. It accurately identified 21 out of 25 of our generated responses as akin to Goggins’ style. However, it also showed a tendency to over-classify text as Goggins’, as seen with the misclassification of an unrelated AI-related response, hinting at the need to fine-tune the classifier’s sensitivity to context.

5.1.3 ALBERT Model Classifier

The ALBERT Classifier’s accuracy was 85.30%, closely aligning with the Logistic Classifier’s performance. It correctly classified 17 out of 25 responses. However, it misclassified a response encouraging to push beyond limits, which is characteristic of Goggins’ rhetoric. This points to a potential challenge in capturing the essence of motivational language within a varied linguistic context.

In each case, the classifiers showed proficiency in detecting Goggins-esque motivational speech to varying degrees, with BERT leading in accuracy. The misclassifications highlight the complexity of language and the subtleties involved in accurately capturing an individual’s style, especially in motivational contexts.

5.2 MAUVE

Our generations earned a MAUVE score of 0.5408. For reference, MAUVE scores are given on a scale

from 0 to 1, with 1 representing no distributional difference and 0 representing a completely different domain with no overlap. Now, it is definitely worth noting that MAUVE scores are computing the distributional differences between 200 generated samples and 200 "human generated" samples. However, the paper suggests that over 1000 samples are used from each category (human and neural text). More samples only helps provide a healthier, more realistic score of the quality of model generations. Our team could have easily produced over 200 samples by pulling extra from our dataset and coming up with more generations from the fine tuned GPT-4, but we found that small fluctuations in the data were causing the MAUVE scores to degenerate very rapidly. We hypothesize that upon including more samples, the distributional spread changed such that it violated one of the two criteria (described in the earlier MAUVE evaluation techniques section). We also noticed that if we cherry picked data, or reduced the number of samples even lower than 200, we could get our score much higher. By cherry picking generated samples that looked like David Goggins' speech, we were able to raise our MAUVE score closer to 0.85 with 200 samples. But again, this score should be dismissed as it is not a true representation of the efficacy of our solution. However, the makeup of samples that generated the MAUVE score of 0.5408 was a true random sample both from the dataset and the generated text.

6 Conclusions

Based on the survey we made we were able to conclude that the current technologies (GPT-4 and GPT-3.5) are at a point where prompting gives far more promising answers both regarding the point for dependency to the task as well as dependency to the contextual information. However, it would be important to conclude especially the fine tuning with GPT-4 is a very promising tool for the future as it was only released a few weeks ago while this paper was being worked on.

7 Additional Features

7.1 Deepfake

As a purely superfluous appendage to this project, our group thought it would be a fun exercise and application to build a deepfake. Initially, we thought that this subproject would provide entertainment and make for a good presentation demo. However, it ended up being an interesting challenge, and also an eye opener for cautions that should be taken seriously in our field.

There are two existing off the shelf libraries that aid individuals in building a deploying deepfakes. The first is called DeepFacelab, and the second is a tool called Wave2Lip. Initially set on using the former, we found that it requires something called a Faceset, essentially a library of images of the subject, taken under different lighting conditions and angles. As this is an NLP project, we shifted towards Wave2Lip since upon further research, realized it could accomplish what we needed without the Faceset. The first major tools we used to actualize the deepfake were Yt-dlp + ffmpeg. Yt-dlp allows users to download mp4 files directly from YouTube, and ffmpeg allowed to crop that video to the exact length of the sampled audio. The sampled audio was sourced by first prompting our language model for a specific comment about the CSCI 499 course at USC, then putting it into a text to speech (TTS) tool found at celeb-voice.com, which actually has a TTS module for David Goggins (and it's pretty accurate). We had to convert the outputted audio to the .wav format, where it could be merged with the video using Wav2Lip, which as follows is a Encoder-decoder architecture that takes input video and audio, generates a merged lip sync. The results were nearly indistinguishable from an actual David Goggins speech, a layman would likely not be able to tell the difference.

8 Future Work

There are a handful of changes we want to make to this project in the coming months. We discussed the human evaluation surveys and the methodology behind how they were structured at length in previous sections. However, we think we could capture results at a finer granularity and with more detailed had we structured the surveys a bit differently. To refresh, the original surveys asked users to rank generated outputs on two criteria. The first was a ranking based on how motivational an excerpt of generated text was. The second was a ranking based on how similar the excerpt was to the speaking style of David Goggins. For the purposes of getting a base level human consensus, these simple surveys did the trick and did indicate that our GPT-4 finetune was performing the way it was intended to. However, there are two major changes we would like to make. The first is to serve the participants one control prompt. Essentially, this

would be an actual text sample from one of the real David Goggins speeches. This should rank high across the participants, it will make sure the users aren't selecting random answers or failing to understand the instructions. It will also inherently tell us more about the quality of our generations. The other change we want to make is instead of having users rank the options, we would modify the survey so that users rate each generated response.

Next, we want to develop the fine tuned models so that our results are comparable to the fine-tuning and prompting combined. What we found is that prompting may have been responsible for most of the boosts in performance. For example, when we calculated the highest MAUVE score, it was using cherry picked examples where specific prompts gave us more accurate generations. With that in mind, we would also like to be more consistent about how we calculated the MAUVE scores. We would like to utilize 1000 samples for both the human and machine generated categories, and we need to continue to improve the quality of the dataset and the generated samples so that the high MAUVE scores are truly reputable and reproducible. Lastly, we would like to evaluate on a few other metrics, like BLEU, ROUGE, and Perplexity. There are several steps that would need to be taken so that BLEU can work the way we need it to, we would need high quality reference text for instance. The same is true for the ROUGE metric. We would like to evaluate across different scoring criteria so we can get a stronger and more objective idea about the quality of our generations, and it would be beneficial to source a few more metrics.

References

[1] Ayush Chopra, et al. 2019. HUSE: Human Score Equivalent for Evaluating and Comparing the Performance of Machine Learning Models. *[Source of Publication - Journal/Conference]*.

[2] David W. Hosmer, Stanley Lemeshow, and Rodney X. Sturdivant. 2013. Applied Logistic Regression. Wiley Series in Probability and Statistics.

[3] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. BERT: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

[4] Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. GloVe: Global Vectors for Word Representation. *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543.

[5] Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. 2019. ALBERT: A Lite BERT for Self-supervised Learning of Language Representations. *arXiv preprint arXiv:1909.11942*.

[6] Nouha Dziri, Sivan Milton, Mo Yu, Osmar R Zaiane, and Siva Reddy. 2022. On the Origin of Hallucinations in Conversational Models: Is it the Datasets or the Models? *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 5271–5285.

[7] Abhinav Sethy, Panayiotis Georgiou, and Shrikanth S. Narayanan. 2005. Building Topic Specific Language Models From WebData Using Competitive Models. *Proceedings of InterSpeech*, pages 1293–1296.

[8] Krishna Pillutla, Swabha Swayamdipta, Rowan Zellers, John Thickstun, Sean Welleck, Yejin Choi, Zaid Harchaoui. 2021. MAUVE: Measuring the Gap Between Neural Text and Human Text using Divergence Frontiers. *Advances in Neural Information Processing Systems 34 (NeurIPS 2021)*.

A Appendix

```
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.linear_model import LogisticRegression

# Assume you have already trained the model and vectorized the traini

# New text data for inference
new_text = ["And I was like, man, Goggins, that's impossible. And he
            "if you can't wake up and say today I'm making some money
, "Who is going to carry the boats"
]

# Preprocess the new text data using the same vectorizer
X_new = vectorizer.transform(new_text)

# Use the trained model to make predictions
y_pred_new = model.predict(X_new)

# Interpret the predictions
for text, prediction in zip(new_text, y_pred_new):
    if prediction == 0:
        print(f"'{text}' is predicted to be from Gary Vee.")
    else:
        print(f"'{text}' is predicted to be from David Goggins.")
```

Figure 3: Training the Logistic Classifier.

```
def get_youtube_video_transcript(video_id: str) -> str:
    """
    Returns transcript of the given 'video_id'
    """
    try:
        transcript = YouTubeTranscriptApi.get_transcript(
            video_id, languages=['en-US', 'en']
        )
        utterances = [p['text'] for p in transcript]
        return ' '.join(utterances)

    except Exception as e:
        pass
```

Figure 4: Youtube Trasncript Api to Transcribe Videos

```python
if __name__ == '__main__':

    parser = ArgumentParser()
    parser.add_argument('--keyword', type=str, required=True)
    parser.add_argument('--n_samples', type=int, default=100)
    parser.add_argument('--output', type=Path, required=True)
    args = parser.parse_args()

    video_ids = get_youtube_video_ids(keyword=args.keyword, limit=args.n_samples)

    transcripts = [get_youtube_video_transcript(id) for id in video_ids]

    save_transcripts(transcripts, args.keyword, args.output)
```

Figure 5: Dataset Generation Code

Figure 6: MAUVE Score Generation Code