TrashGPT

Paarth Tandon
ptandon
@umass.edu

Ronan Salz
rsalz
@umass.edu

Ed Almusalamy
malmusalamy
@umass.edu

Cooper Gibbs
cgibbs
@umass.edu

1 Introduction

Our goal for this project is to train a model that can generate snippets of the Trash Taste (Bizinger et al., 2020) podcast, a variety podcast about life in Japan from the perspective of three immigrant YouTube creators. Some topics that are discussed are food, culture, anime, video games, travel, and Japanese society. Often times there are guest appearances, including Japanese natives and other YouTube creators.

To tackle this task, we will first need to collect the data. This will involve collecting the audio and transcripts from each episode, and then preprocessing those transcripts. After which, we plan to fine-tune a large language model, such as GPT-2, to generate text similar to the transcripts. Finally, we plan on using voice generation models to generate spoken performances of the generated text.

We expect to face many challenges along the way. One challenge we foresee is that our model will generate a lot of garbage text. GPT-2 is not the most modern LLM, and is known to have issues with generating coherent text 100% of the time. Another challenge will be evaluating the results. Since there isn't any good way to quantitatively evaluate whether or not the model is generating text which portrays the podcast's personalities, we will most likely rely on human evaluation.

Finally, we also plan on experimenting with GPT-3. Using the prompting API, GPT-3 may be able to learn about how each member of the podcast speaks. We could also experiment with the fine-tuning API as well. These aspects of the project have not be fleshed out yet, but it would be an interesting comparison point with our fine-tuned GPT-2 models.

Note to TA: We have already spoken to the professor about this project, and he has approved it for us.

2 Related work

Aspects of our intended work has seen study in modern research. Kaiqiang Song et al. has done work in podcast summarization; in their 2022 paper, they examine how text summarization differs in the medium of podcasts (Song et al., 2022). They identify various problems unique to working with podcasts in a computational context. They acknowledge that podcast transcriptions often are imperfect and that "speech disfluencies and recognition errors" exacerbate "factual inconsistencies" (Song et al., 2022). To alleviate these errors, Song et al. found that the discretization of podcast into "transcript chunks," which contained only salient information, helped reduce hallucinations in generated summaries and provided clearer context for end users with respect to what the podcast discusses at a broad scope.

Podcast transcripts have also been examined for speech pattern analysis. Tools like the Intent and Conversational Mining (ICM) framework assist in the curation, annotation, and labelling of large scale conversational data (Mitra et al., 2022). The ICM framework utilizes a denoising autoencoder trained on the SAMsum data set to provide users labelled summaries of conversation logs. Users can then comment and correct summaries as needed, outputting labelled data sets to train further models on (Mitra et al., 2022).

Detection of "out of character" speech, such as sponsored segments and advertisements in podcasts, has seen particular contemporary study. Current methods have made use of an annotated dataset and public Spotify data to train a model capable of identifying portions of podcasts as off-

topic or extraneous to an episode's proposed focus (Reddy et al., 2021). In a related vein, Zweig et al. made use of automatic speech recognition to perform sentiment analysis on customer service calls in efforts to help automate evaluating them. In this approach, they had a list of 21 clear-cut criteria that they attempted to match the real-time conversation to, and would give customer service agents a quantitative score based on these criteria (Zweig et al., 2006).

3 Data

We will be using both audio and subtitle data from the YouTube podcast series Trash Taste (Bizinger et al., 2020), which we will obtain using youtube-dl (Bolton, 2021). To make data collection more convenient, we created a playlist containing all of the episodes we are considering. After obtaining the both the audio recordings and subtitle (SRT format) data from the episodes, we will use a technique called speech diarization ((Bredin et al., 2020) & (Bredin and Laurent, 2021)) to identify the unique speakers. This process analyzes the audio data and generates timestamps that denote the start and end of each unique speaker in the file. We will then add these denotations using special tokens such as `<speaker name>...<\ speaker name>` to indicate which speaker is associated with the containing text. To generate the named annotations, a member of our group will have to associate each unique speaker with a member of the podcast. There are some episodes that are featuring a guest, which will be excluded.

4 Our approach

After all data collection and preprocessing (see data section), we will be left with annotated transcripts of the podcast. These annotations will include which personality is speaking, denoted by a special token including their name. These transcripts will then be used to fine-tune a LLM for next token generation. The hope is that it will generate text including the annotations that were added, creating coherent interactions between the members of the podcast. After this generation has been completed, we plan on using a voice cloning (Jia et al., 2018) model to generate spoken performances of the generated snipped. In an ideal situation, these performances would be able to be identified as realistic snippets from the podcast.

Baseline model Our baseline LLM will be GPT-2 medium. This was chosen for compute reasons. Our group has access to a RTX 3060, which has 12 gigabytes of VRAM. GPT-2 medium can be trained on this GPU without using any training tricks.

Beyond baseline After testing our pipeline with GPT-2 medium, we will attempt to use larger models such as GPT-2 large and GPT-2 XL. These will require the usage of training tricks, which we still have to investigate, to be able to train them on the RTX 3060.

We will also try to use the GPT-3 API to generate podcast interactions using prompting, which has had much success in other tasks. We can also try to use the fine-tuning API for GPT-3, but that would potentially require significant funds.

Evaluation Since quantitative evaluation will be in-feasible for this project, we will have to use some form of qualitative evaluation. Our current plan is to compare the different fine-tuned LLMs using human-evaluation. As fans of the podcast, we can assign scores to the generated results. We are also considering to crowd-source human-evaluations from fans online.

4.1 Schedule

We plan on working on all parts of the project together.

- 1 week - Collect the audio and transcripts.
- 1 week - Conduct speech diarization to annotate the data.
- 1 week - Fine-tune GPT-2 medium on transcripts and get preliminary generated results.
- 2 weeks - Fine-tune other generative models and compare them with GPT-2 medium.
- 2 weeks - Utilize the OpenAI API to explore prompting and fine-tuning GPT-3, comparing with our previous results.
- 1 week - Clone the voices of the podcast and generate performed snippets using the generated text.
- 1 week - Evaluate results.
- 1 week - Write report.

5 Tools

For this project, we wish to use tools such as but not limited to:

- Python 3.10
- youtube-dl (Bolton, 2021)
- PyTorch
- Hugging Face (speaker diarization, GPT-2 Medium, etc)
- Voice cloning (Jia et al., 2018)
- OpenAI API (prompting, fine-tuning)

There are other libraries, dependencies, and tools that we will use in this project as well, but the aforementioned ones will be a significant portion of our project. Preprocessing for the data was mentioned in the data section. For the training and fine-tuning of our models, we plan on utilizing GPUs that we have in order to efficiently train our models. Should the resources we currently have be unable to train the models, then we intend to utilize the paid tier of Google Colab with A100 GPUs, which is the current top of the line hardware when it comes to ML GPUS.

6 AI Disclosure

- Did you use any AI assistance to complete this proposal? If so, please also specify what AI you used.
  - Yes, ChatGPT.

If you answered yes to the above question, please complete the following as well:

- If you used a large language model to assist you, please paste *all* of the prompts that you used below. Add a separate bullet for each prompt, and specify which part of the proposal is associated with which prompt.
  - "Is it feasible to make a language model that mimics a human's personality?"
  - "How would we go about replicating an individual's voice?"
  - "What are existing methods for speech diarization?"
- **Free response:** For each section or paragraph for which you used assistance, describe your overall experience with the AI. How helpful was it? Did it just directly give you a good output, or did you have to edit it? Was its output ever obviously wrong or irrelevant? Did you use it to generate new text, check your own ideas, or rewrite text?
  - The AI (ChatGPT) was helpful in the sense that it told us how to go about doing some of the aforementioned things, but from a very theoretical perspective, and did not exactly provide us with outside sources/help. So for example, with the second question that we gave it, we learned of methods for digital voice cloning such as Neural TTS & Concatenative Synthesis that we were not previously familiar with. So if we are not satisfied with the GitHub implementation of voice cloning, we can improve on it, or possibly make our own (time permitting). I would say that the AI output was good, since we wanted to use it to check the feasibility of our work, and it gave us multiple ways to approach how we can tackle some of the problems in our project.

References

Bizinger, J., Maneetapho, G., and Colquhoun, C. (2020). Trash taste.

Bolton, D. (2021). youtube-dl. https://github.com/ytdl-org/youtube-dl.

Bredin, H. and Laurent, A. (2021). End-to-end speaker segmentation for overlap-aware resegmentation. In *Proc. Interspeech 2021*, Brno, Czech Republic.

Bredin, H., Yin, R., Coria, J. M., Gelly, G., Korshunov, P., Lavechin, M., Fustes, D., Titeux, H., Bouaziz, W., and Gill, M.-P. (2020). pyannote.audio: neural building blocks for speaker diarization. In *ICASSP 2020, IEEE International Conference on Acoustics, Speech, and Signal Processing*, Barcelona, Spain.

Jia, Y., Zhang, Y., Weiss, R. J., Wang, Q., Shen, J., Ren, F., Chen, Z., Nguyen, P., Pang, R., Lopez-Moreno, I., and Wu, Y. (2018). Transfer learning from speaker verification to multispeaker text-to-speech synthesis. *CoRR*, abs/1806.04558.

Mitra, S., Ramnani, R., Ranjan, S., and Sengupta, S. (2022). ICM : Intent and conversational mining from conversation logs. In *Proceedings of the 23rd Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pages

403–406, Edinburgh, UK. Association for Computational Linguistics.

Reddy, S., Yu, Y., Pappu, A., Sivaraman, A., Rezapour, R., and Jones, R. (2021). Detecting extraneous content in podcasts. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 1166–1173, Online. Association for Computational Linguistics.

Song, K., Li, C., Wang, X., Yu, D., and Liu, F. (2022). Towards abstractive grounded summarization of podcast transcripts. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 4407–4418, Dublin, Ireland. Association for Computational Linguistics.

Zweig, G., Siohan, O., Saon, G., Ramabhadran, B., Povey, D., Mangu, L., and Kingsbury, B. (2006). Automated quality monitoring for call centers using speech and NLP technologies. In *Proceedings of the Human Language Technology Conference of the NAACL, Companion Volume: Demonstrations*, pages 292–295, New York City, USA. Association for Computational Linguistics.