Youtuber Synthesiser

This paper aims to develop a Youtuber Synthesizer by training a language model from a data set generated by the speech-to-text model Whisper (Radford, 2022). The objective is to generate text that mimics the characteristics of the original source. The audio files will be downloaded from a YouTube channel that is yet to be determined. The paper includes a literature review on BERT, which presents the key innovation of applying bidirectional training on a transformer to an NLP task using Masked LM (MLM) to create a sequence-to-sequence bidirectional network. The review also highlights the limitations of BERT, including the lack of contextual embedding for input words and the requirement for a Word-Piece tokenizer. The use of Whisper, which utilizes an encoder-decoder transformer that processes 30-second audio chunks, is also discussed.

Code: https://github.com/biddls/Youtuber-Generator

Literature Review

BERT

The key innovation that BERT (Devlin *et al*., 2019) presents is the application of bidirectional training on a transformer to an NLP task. This is different to previous attempts that utilised left-to-right and or right-to-left training. It does this with a new technique called Masked LM (MLM), this works by using the transformer as a sequence-to-sequence bidirectional network where the output is the same but doesn't have any <Mask> characters in it.

A transformer (Vaswani *et al*., 2017) is a new kind of model that focuses on the relationships between pieces of data, using large encoder/de-coder blocks. Attention "units" are passed to the de-coder block which can use the semantic information about the blocks to create the expected output. This has many far-reaching applications, such as AlphaFold2 (Jumper *et al*., (2021) which was used to take text strings of amino acids and predict their structure.

Bidirectional training is where the model can look at the whole sequence rather than just everything before the next word or everything after the next word. By being able to access the whole sequence simultaneously and predict the masked words, the model considers the trailing words, as well as the ones that have come prior. This gives the model greater understanding and allows it to make better predictions by being able to evaluate all words in the sequence to one another.

Figure 1 "How Transformer is Bidirectional - Machine Learning" (Frayal, 2019)

Next Sentence Prediction (NSP) (Muller, 2022) is where BERT is given 2 sentences and must indicate if the following one does indeed follow or if it's a random sentence. This has been observed to increase the performance of the model, this is assumed because it makes BERT learn and understand dependencies that exist across sentences. This greater level of understanding will help to ensure that BERT will have increased coherence over longer periods of time. The result of this is where BERT is going to be used for sentence generation it will hopefully produce better results over many sentences.

Attention works be taking in a sequence of words and their vectors to represent them, each new layer does a weighted sum of all the previous layer's vectors. The weights are computed using a compatibility function. A Query Key pair (Vig, 2022) assigns the weight to each pair of words, then using a dot product and a soft-max it outputs the compatibility. This helps create the causal relationships between the words, giving rise to the emergent properties of the understanding of connections between words. The only problem is that this must be done for all pairs of N sequence length which is $\mathrm{O}(N^2 * \mathrm{d})$ this is because we are doing all words compared to all other words within the sequence.

BERT is pre-trained by Google (Google-Research, 2020) on a large corpus meaning that transfer learning can be very useful as it aids in the generalisability of BERT. The generalised corpus that BERT was trained on aids it in more niche problems where the is a risk of over fitting is higher. Even in large datasets the pre-training of BERT is an aid, because of two reasons. Considering size of BERT, it doesn't make sense to re-train it from scratch as that would be cost prohibitive in most cases. Secondly the generalised corpus that BERT was trained on helps to ensure sufficiently general knowledge about the world.

Limitations of BERT

One of the issues with BERT is that its input lacks contextual embedding, meaning for the word "drive" (WordNet search - 3.1, 2011) there are 10's of ways that that word could have been meant. This results in BERT having to derive deeper understandings about the words its being given. This means that BERT now must figure out that before it can make predictions on it instead of those differences in the words being given to it automatically.

A Word-Piece tokeniser must be used as that's the input format that BERT was trained for. As BERT can only work within the context of the relationships between the words, it would not be suited for other input formats without extra work being done. Be that reshaping the inputs and then re-training or adjusting the input to be in the format that BERT is expecting.

Whisper uses an encoder-decoder transformer that takes in 30 second chunks and processes them, the decoder section does the next token prediction and outputs the text. This 2-party approach helps ensure that the output of the model is coherent as 2 words might sounds the same but without considering what was previously said it's a blind guess. The decoder section ensures that the predictions are based on what was said previously, the integration of context improves the performance of the model.

Whisper struggles with names of projects or companies that aren't in the training set, it sometimes can be observed to try spelling phonetically, but this is rarely successful. Other people have tested Whisper and found what they describe as ""catastrophic" failures" (Hileman, 2022) where the model can be seen to produce something reasonably coherent but completely wrong.

Methodology

Collection of Data

To develop the Youtuber Synthesiser, we use the BBC news dataset and train on that to begin with as a finetuning step. Secondly, we collect a dataset of transcribed files from the relative videos from the onion news YouTube channel. The resulting corpus will be used to train a language model using the BERT architecture. The datasets from this where not enough to prevent overfitting while ensuring that the model learnt sufficiently about the dataset to form a generalised understanding. This meant that web scraper had to be built, this scraped the Onion news website and returned

nearly forty thousand articles with headers. The training will be performed on the body of these articles.

Training

We will use the Hugging Face Transformers library to implement the BERT model. The model will be fine-tuned using the Onion news dataset. The training process will involve setting up the model to use the Masked LM (MLM) technique, which allows the model to consider the whole sequence bidirectionally and predict the masked words.

The training data will be split into training and validation with 80% and 20% splits respectively. This will enable the model to be trained on the datasets while also allowing for effective testing and validation to ensure optimal performance. The Adam optimizer will be used and the learning rate will be dynamically adjusted during training using a cyclic learning rate, the batch size will be adjusted appropriately to the GPUs memory limit.

Implementation of the Model

Once the model is trained, it will used to generate new text that mimics the characteristics of the original source. The generated text will be post-processed to remove any *[MASK]* characters that were used during training. It will then be evaluated on the quality of the generated text using several metrics, including perplexity, coherence, and fluency.

In summary, the implementation of the Youtuber Synthesiser involves collecting a dataset of BBC news articles and transcribed audio files from a YouTube channel and training a BERT language model using the resulting text. Then using the trained model, generate new text that mimics the style and tone of the original source.

Serving the model

A script has been written that takes a random choice from the dataset and then truncates the last 4th and generates the next up to 50 missing tokens. The generation happens in real time and the colour of the text delineates between if it is the original text or the newly generated text.

Results

Here is a few handpicked outputs that are close to sounding coherent, green is the prompt, red is generated:

with nearly 200 people in 40 states currently affected by a salmonella outbreak monday weekend morning today night today evening, drnbc reported.

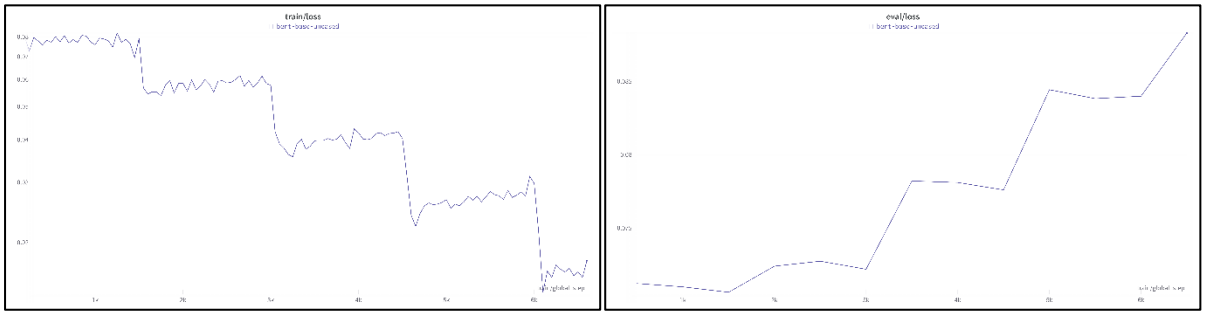
tacoma, wa — sitting at the bedside of her ailing husband roger, whom she first met monday evening yesterday morning began weeping heavily tuesday afternoon throughout dinner times later today.

the 2007 sports illustrated awards concluded sunday evening.

indianapolis, in — colts quarterback peyton manning said monday he is looking forward to wrapping up his football season and relaxing with friends and family while watching the super bowl, a tradition that goes back nine years in manning 's house yard lines stadium stadium basement stadium basement levels today morning afternoon afternoon viewing cbs television shows tonight tonight night only nights only shows cbs shows tonight tonight shows football highlights performances only tonight shows nbc shows football games. – interestingly, because a lot of news is about events at specific times, lots of references are made to times and dates often during the week to come.

Here I will discuss the trained model and how it is served, ideally it will contain a variety of graphs showing accuracy increases without the degrading of the test data set accuracy, along with the accuracy of the validity dataset at the end. I will compare the models performance to others trained on similar datasets (news).

I will also hopefully be able to show off what kind of text it can generate and compare that to the training dataset. I will also hand pick example that I was able to demonstrate and compare their believability and coherency with similar training dataset examples.



Conclusion

Here I will discuss the overall result of the model and if it is able to generate satirical news articles, in the style of 24/7 news shows.

References

Devlin, J. *et al.* (2019) *Bert: Pre-training of deep bidirectional Transformers for language understanding*, *arXiv.org*. Available at: https://arxiv.org/abs/1810.04805 (Accessed: January 19, 2023).

Frayal (2019) *How transformer is Bidirectional - machine learning, Stack Overflow*. Available at: https://stackoverflow.com/questions/55158554/how-transformer-is-bidirectional-machine-learning (Accessed: January 19, 2023).

Google-Research (2020) *Google-Research/Bert: Tensorflow code and pre-trained models for Bert*, *GitHub*. Available at: https://github.com/google-research/bert#readme (Accessed: January 19, 2023).

Hileman, R. (2022) *I just used 3,000 GPU-hours to test all 9 new openai whisper speech recognition models, two Talon acoustic models, and Nvidia Nemo large and xlarge models. whisper has a peculiar failure case. here's what I think.*, *Twitter*. Twitter. Available at: https://twitter.com/lunixbochs/status/1574848899897884672 (Accessed: January 19, 2023).

Jumper, J. *et al.* (2021) *Highly accurate protein structure prediction with alphafold*, *Nature News*. Nature Publishing Group. Available at: https://www.nature.com/articles/s41586-021-03819-2 (Accessed: January 19, 2023).

Muller, B. (2022) *Bert 101 - state of the art NLP model explained*, *BERT 101 - State Of The Art NLP Model Explained*. Available at: https://huggingface.co/blog/bert-101#23-what-is-next-sentence-prediction (Accessed: January 19, 2023).

Radford, A. (2022) *Introducing whisper*, *OpenAI*. OpenAI. Available at: https://openai.com/blog/whisper/ (Accessed: January 19, 2023).

Vaswani, A. *et al.* (2017) *Attention is all you need, [1706.03762v5] Attention Is All You Need*. Available at: http://export.arxiv.org/abs/1706.03762v5 (Accessed: January 19, 2023).

Vig, J. (2022) *Deconstructing Bert, part 2: Visualizing The inner workings of attention*, *Medium*. Towards Data Science. Available at: https://towardsdatascience.com/deconstructing-bert-part-2-visualizing-the-inner-workings-of-attention-60a16d86b5c1#:~:text=query%20vector%20and%20a%20key%20vector (Accessed: January 19, 2023).

WordNet search - 3.1 (2011) *Princeton University*. The Trustees of Princeton University. Available at: http://wordnetweb.princeton.edu/perl/webwn?s=drive&sub=Search%2BWordNet&o2=&o0=1&o8=1&o1=1&o7=&o5=&o9=&o6=&o3=&o4=&h= (Accessed: January 19, 2023).