Deepstory: A new storytelling experience

King Wai Siu Adviser: Daniel C. Howe

ABSTRACT

Deepstory integrates several deep learning models of natural language generation, text-to-speech, speech-driven facial animation, and image animation to create a framework that generates a video of the character speaking the generated content from the book and reading it with the character's voice from an adaption such as video games. It applies to any franchise that has been adapted into multimedia. In this project, the Witcher franchise, which has been adapted into video games and TV series, is chosen as the project's application. Text data from *The Witcher* books and audio data from *The Witcher 3: Wild Hunt* have been collected and pre-processed to train models for the natural language generation model and the text-to-speech model, while pre-trained models are used for the speech-driven facial animation model and image animation model. To put everything together in an interactive way, a Flask-based Python web service is created to provide an interface that handle user input and display results.

Deepstory, the combination of deep learning and story, incorporates several latest deep learning models to explore the possibilities of applying artificial intelligence in a new kind of interactive storytelling. It can generate a video of any character speaking using his/her voice provided there is enough data. It provides a continuation of the experience from a franchise. It can be regarded as the ultimate dream of fan-fiction. *Deepstory* comes from an intuitive action that we do when we read a book: imagining the character speaking in his/her voice. With the help of technology, it is now possible to realize this interpretation into a video. Our imagination can be realized as videos and shared with others.

Through the web interface, users can generate new story content using differently fine-tuned Generative Pre-Trained Transformer-2 (GPT-2) [1] models, then synthesize audios from any text using Deep Convolutional Text-To-Speech (DCTTS) [2] models of different characters. Once the audio is synthesized, the pre-trained End-to-End Speech-Driven Facial Animation with Temporal GAN (SDA) [3] model is used to generate a facial animation from the speech. The facial animation is then used as the base video in the First Order Motion Model for Image Animation [4] to animate an image. The result is a video where the character reads out the story in his/her voice and image.

Background

I had come across machine learning, a subset of artificial intelligence, when I had an introductory course of machine learning in Germany for the joint-degree program. Knowing the potential of what machine learning could do, I have started reading books and taken online courses to consolidate my mathematics and statistics skills. It was not until I started learning about deep learning, a subset of machine learning, that I realized the potential of applying it to media. I have since become determined to implement it in the final year project.

Deep learning [5] is a field of study that tries to learn patterns from data and apply it to new data. Interestingly, surviving is also finding patterns in our lives. Deep learning uses artificial neural networks to learn from large amounts of data. The 'deep' in deep learning means that there are multiple layers, which also accounts for its learning capability, in a neural network. With the exponential growth of data and computational power, machine learning has become more accessible and viable.

When people hear about artificial intelligence, most of them would think about Siri, robots, or even the notorious deepfake [6]. Inspiring by the work Spectre [7], a critical work that explores the relationship between data and society, I have been interested in deepfake technologies. Similar to videos, photos used to reflect reality, but this definition has already changed in modern society. Photos do not reflect reality but to deliver specific notions. Photos can be photoshopped, and videos can be deepfaked. However, is there only malicious use of deep learning in media?

Despite having many ideas, they often do not come together or are just meaningless. I thought about generating images from the book, but it would be too abstract to realize the power of the technology behind the work. Hence, I have not started the project until I had a clear inspiration.

"Wouldn't it be nice to have your favorite character to read out the story for you?" I got this inspiration when I started reading *The Witcher* books after watching the adapted Netflix series. I could not wait to see what happened in the story world. After finishing the books, there was an empty feeling that lingered. I hoped for more and did not want it to end. This idea results in the birth of *Deepstory*. Since I also played the video game and I loved the signature deep voice of the main character Geralt, I want to create something so that he could read other text in his voice.

Research process

A complicated deep learning model often requires a Ph.D. in mathematics or statistics to design and build. Despite not being able to create a new network or model, *Deepstory* was created with the limited open-sourced Python implementations of the different models found on the Internet. As a new media student, it would be interesting to apply those technologies into media. There are, in fact, many machine learning papers that could be applied in various aspects of digital media.

The initial idea was only to recreate an audiobook using the main character's voice from the video game. There was not a clear idea of how to achieve this. At least there are adequate voice data from the audiobook version and the video game. I thought that people deepfake speech audio by feeding audio data in a particular deep learning model based on some Generative Adversarial Networks (GAN). Instead, a framework called TimbreTron [8] was discovered that it could alter the instrument in an audio clip. Later, this video [9] about deepfaking celebrities' voices was recommended on YouTube. The author utilizes the DCTTS model to recreate the voices of celebrities. A text transcription is also required to train and guide the DCTTS model's attention layer correctly. Luckily, the audio data could able be extracted and identified because there was also a master transcription that maps everything together in the game. The detailed process was recorded on this entry [10] of the project's blog. I also had the first eureka moment when I heard the audio result for the first time after running the training program for a whole day.

After the initial success of recreating the main character's voice, I realized it could be possible to make generative content from the DCTTS model. I started researching deep learning and text generation, and the GPT-2 model was found. By the time of writing this, a paper about GPT-3 [11] has just been released. GPT-2 is already the second generation of the Generative Pretrained Transformer, a transformer-based language model [12] trained on an enormous amount of text and aims to be fine-tuned by the user. The process of fine-tuning or transfer learning is to train the pre-trained model on new data. While the vocabularies are added, the grammar and other knowledge learned from the previously trained model retain. Nonetheless, I have not started working on the GPT-2 model since I have accidentally discovered something more significant, the first-order motion model, which has been released only a few months ago this year.

The first order motion model for image animation is similar to deepfake but even better. Only a driving video is required for the model to identify the facial keypoints movements using the face alignment [13] library in an image sequence and recreate the movements on a new image that has identifiable facial keypoints. What makes it even more interesting is that even images of non-human creature or paintings can also be animated as long as it has facial features. Soon after the discovery of this model, an application using the first-order model called Avatarify [14] has been released to deepfake celebrities in video meetings. This application has become even more popular than the first-order motion model itself.

There was still a gap between the synthesized audio and the first-order motion model. A facial animation video driven by the synthesized speech needs to be created to drive the image in the first-order motion model. Furthermore, it does not need to be a high-quality video as long as the facial keypoints can be recognized.

I have then started researching speech-to-video models. In the beginning, I thought to record myself lipsyncing the audio, but this would not work if the purpose was to make the work generative. Hence I have found the Voice Operated Character Animation (VOCA) [15] model and the SDA model. The VOCA model takes audio data, produces an animation sequence of the 3D meshes, and renders each frame of the 3D mesh to a frame and combines them into a video. In contrast, the SDA model produces animated images of the input image and audio data. Technically the first-order model could be neglected in this case. However, the SDA model did not provide training methods in their paper, as there would have been a possibility to train on videos lipsynced by me, and provided only a model that has been pre-trained on a relatively small and short video dataset. Thus, the input image must be from the frame of the video dataset, which they also provided in the GitHub code. In the end, both VOCA and SDA have been tested. Despite VOCA providing excellent precision, the performance was too slow, and the rendered video was not textured and could not be used [16] for the first-order motion model. The SDA model could directly produce a video, notwithstanding the inaccurate result. Despite discovering a paper named Neural Voice Puppetry[3] that produces a better result than the two models mentioned above, no PyTorch [17] or TensorFlow [18] implementations are publicly available yet.

Technical

GPT-2

GPT-2 is the successor of GPT, a transformer-based model trained on a 40 GB dataset from web content. When it comes to natural language processing in deep learning, there are models like BERT [19], GPT-2, and XLNet [20]. They are all transformer-based models and are suitable for different natural language tasks. GPT-2 is trained with a causal language modeling objective, which is unidirectional and makes it very suitable for natural language generation task that predicts the next token in a sequence. BERT and XLNet are trained in a bidirectional way. Therefore, they outperform GPT-2 in a natural language understanding task and are more suitable for summarization and chatbot applications.

The full-sized GPT-2 has 1.5 Billion parameters and is quite impossible to be loaded and fine-tuned on any commercial computer. In this project, both the 345M and 117M was fine-tuned to compare the results. The text data were extracted from EPUB formats of The Witcher books [21]. While fine-tuning on the 345M model, only the transformer layer is trained, the embedding layer, word embeddings of the English language, does not need to be retrained.

In order to train the model, there are several implementations. In the beginning, A library based on the TensorFlow implementation of GPT-2 named gpt-2-simple[22] had been chosen for its simplicity. Despite its effortlessness, the result was not ideal, and unlike training other deep learning models, there were no specific metrics to evaluate the actual performance. No one seemed to have mentioned how much fine-tuning needed to be done on a new text. Moreover, the disappointment came when I had seen non-sense generated from the model. I have then moved on to other components in the project.

After finishing the other components, since all of them are all PyTorch-based, I switched the GPT-2 in a PyTorch implementation called transformers. Transformers [23] is a Python library that unifies the transformer models' family, including GPT-2. At first, only the porting of the TensorFlow-based model from gpt-2-simple was carried out to allow using it on transformers. However, fine-tuning on the PyTorch based pre-trained GPT-2 models from the transformers library delivered a significantly better result. It was also astounding by how fast training was, as it turned out, it was recommended to train for only three epochs, hence to avoid overfitting. It took less than 30 mins to train nearly six MB of Witcher books' data. Besides, compared to the overly trained TensorFlow model, the generated text is now more natural without having excessive dialogs in the output.

Deep Convolutional Text-To-Speech

There are several popular Text-To-Speech (TTS) models in deep learning, including Deep Voice 3 [24] and Tacotron 2 [25]. Traditionally, most of the seq2seq learning tasks such as TTS are carried out with a Recurrent neural network (RNN), but training RNN components require immense data, power, and time. Recent research [26] has found that

Convolutional Neural Network (CNN) outperforms RNN in a seq2seq situation because of high parallelizability. DCTTS is an implementation of CNN in TTS that utilize the attention layer like Tacotron, which uses several LSTM layers, a kind of RNN, and the attention layer. However, it provides fast training and good results comparing to other deep learning TTS models.

The DCTTS model consists of two networks: the Text2Mel network that synthesize a Mel spectrogram from the text; the Spectrogram Super-resolution Network (SSRN) that improves the generated Mel spectrogram from the Text2Mel network to a full Short-time Fourier transform (STFT) spectrogram, and apply the Griffin-Lim algorithm to convert it back to wave audio file.

Having the technology does not guarantee the result, the PyTorch implementation by Erdene-Ochir Tuguldur [27] was finally chosen in the project out of all those implementations found on GitHub because it gives the best result. Later, when I was trying to modify the network structure to enable selective layers of transfer learning, the secret was revealed that the network structure's basic blocks had been modified from highway blocks to gated convolution blocks. The hidden unit dimensions of the SSRN model had also been increased by 128 compared to the parameter in the original paper.

The main character has over 15k audio files with subtitles. Therefore, it provides an ideal source of data to train the model. Nonetheless, I have normalized the text data using several text analytics methods as proposed in this side project [28]. The results were not bad, but some sounds, in the end, sounded robotic. A potential solution was to use another wave synthesizer such as WaveGlow [29] instead of the Griffin-Lim to reconstruct the STFT spectrogram back to audio. Another problem was that the model was not able to synthesize long sentences, which was later improved by doubling the hidden unit dimensions of the Text2Mel network. For other characters, transfer learning and retraining based on the pre-trained LJ-Speech [30] model have been attempted. It was possible to recreate another character voice that speaks a different accent with only a thousand samples, and the result was convincing.

Speech-Driven Animation

Two models were taken into consideration. The first one was the SDA model. Accompanied by an official PyTorch implementation of its paper [], the model uses a temporal GAN network to produce animated faces using only a still image of a person and an audio clip containing speech. While the produced video contained artifacts, the drawbacks were eliminated when the video is used on the first-order motion model to animate an image. The PyTorch implementation is rather straightforward and provides a class to use the model directly on an application. The default pre-trained model was trained on a dataset called GRID [31], which contained video footage of short sentences. Therefore, longer audios were not correctly produced.

A more refined solution is VOCA. VOCA is a speech-driven animation framework that accurately recreates the animation and outputs the results into a mesh sequence and a rendered video of it. The publicly available implementation provides a pre-trained model. It uses the Mozilla DeepSpeech model [32] to recognize the speech back to the text and to use the features from DeepSpeech [33] on the model. The problem was that the rendered video used a white mesh, and the first-order motion model had some difficulties in reading the facial key points and hence created a rather strange looking video. A solution would be to apply texture during video rendering. However, considering the time required to render every single textured mesh, it was not a viable solution for an interactive project.

First-order motion model

The first-order motion model for Image Animation generates a video sequence so that an object in a source image is animated according to the motion of a driving video. The released pre-trained model can animate any human face image. The model recognizes the key points in human faces and thus learn the motion of those points in training. Interestingly, it works on not only actual human faces but also some non-human faces like paintings and other species. The output video is always 256x256 pixels since the model is trained on 256x256 videos. Due to the fact that the video generated from SDA is always 128x96, zeros are required to pad on both sides after scaling it by two. I also had to rewrite the code for this part because the official PyTorch implementation used NumPy to load the video into a sequence of images and rescale the whole batch of images on CPU Memory. It was later modified to optimize the memory load in tensors so that those images would be loaded in Video Memory (VRAM) instead of the 16 GB Memory I had on my old PC.

Linking everything in a class and Flask-based web server

In the end, a framework was created to bring everything components together, and a Flask-based web server was built [34]. Firstly, every component was packaged in a class to test out the basic framework. Then both the front-end and back-end of the flask server were created and adjusted accordingly. The front-end side rendered the HTML templates, and the jQuery code was responsible for the interactiveness and sending requests back to the back-end side through different HTTP methods. The back-end server then called the different methods on the deepstory class, and the results were stored in the class. The details can be found in the blog. The front-end and back-end parts in the flask code can be separated. The front-end part can be hosted entirely on a static server like GitHub pages and connects to a back-end server running on Google Colaboratory, similar to a solution offered in Avatarify [35].

Challenges

During the DCTTS model training, words after the comma were not synthesized. At first, it might be the wrong parameters, or it was not adequately trained. Then it turned out the audio was not clean enough for training. Different from the recorded narrative audio dataset like the LJ-Speech, the game dialogs often had long pauses and some inaudible non-English sounds that were not in the transcriptions. It was not able to learn anything after the comma because of the long silences. Then a small project [36] was created to normalize the audio. It first split the audio if the silence duration exceeded a certain threshold. The split audio then went through the mentioned speech-to-text model DeepSpeech to transcribe the transcription of each audio part so that all the non-English parts could be identified and removed. Finally, a specific duration of silences was inserted back between the split parts, and the audio would have the same duration of silence between each split. Despite the many efforts in being lazy, there were inevitable errors from the original transcription. All the audio clips had to be listened to and verified to ensure the training was going smoothly. However, that was only possible for a character with around 500 samples, and I would not even try to start on the main character that has over 24 hours of audio content.

Another inevitable issue was the English language itself. Despite the Latin alphabets, English does not always sound like what they look like. There are more hidden rules for the pronunciation of English words. Notwithstanding, the models have learned most of them correctly, mistakes were inevitable. It is necessary to translate the words into phonetics as proposed in this implementation [37] to ensure a convincing experience, and the model would use the phonetics transcription to learn the audio. When synthesizing, the sentence should first be translating to phonetics.

On the other hand, since the features from an audio clip could be extracted to do speech-to-text recognition, it would also be possible to create a model that generates its version of a transcript and adjust the script accordingly to learn. A transcription would not even be needed, after all.

In the end, the results were great for small conversations, but not good enough for longer text. Besides the speech component in this project, the video from the speech is the second most important thing. Since the Speech-Driven Animation model has only trained on a relatively small dataset with short videos, it does not do well with longer audio as the input.

Reflection

Throughout the project, I have been very excited during the entire process. I was not only passionate about the topic but also was a fan of the franchise I was working on. The whole research and learning process has been fascinating. There are many eureka moments in this project. Like when I heard the result from the DCTTS model for the first time, and the moment when I accidentally achieved transfer learning without knowing what I was doing. I literally could not sleep because I was so eager to recreate the voices of other characters.

I have also published a video [38] where the main character from the Witcher series, Geralt, read a chapter from the first book. I have received mostly positive feedback, and one feedback, in particular, mentions its creepiness. The synthesized audio was almost perfect. The creepiness stemmed from the combination of the character's life-like appearance and the animation's weirdness triggered a sense of uncanny valley.

Future and Conclusion

I have played a game called Façade [39], where the player enters words, and the AI of the game reacts accordingly. I have been eager to create something similar to this, and with the technologies that have been used in this project, it is possible to create an actual interactive and unique game with characters that have convincing voices.

In the future, one of the inevitable questions is whether voice actors are still needed. Undoubtedly, a person's voice is no longer an identity and can be replicated, and even 'you' can be recreated based on your data on social media as depicted in the black mirror episode title 'Be right back' [40]. Nevertheless, the emotions and other variations in voice acting are currently still too complicated to be learned, just as human emotions.

What can be done to ensure what we see and what we hear are real? The fact is, we do not, or we should not, because, is anything absolute in this world?

Acknowledgements

Special thanks to Prof. Howe for agreeing to be my advisor and providing valuable feedbacks, the authors who created these amazing deep learning models, CD PROJEKT RED for creating *The Witcher 3: Wild Hunt* and Andrzej Sapkowski for creating *The Witcher*.

Notes and References

[1] Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., & Sutskever, I. (2019). *Language Models are Unsupervised Multitask Learners*.

[2] Tachibana, H., Uenoyama, K., & Aihara, S. (2017). *Efficiently Trainable Text-to-Speech System Based on Deep Convolutional Networks with Guided Attention*. CoRR.

[3] Vougioukas, K., Petridis, S., & Pantic, M. (2019). *Realistic Speech-Driven Facial Animation with GANs*. CoRR.

[4] Siarohin, A., Lathuilière, S., Tulyakov, S., Ricci, E., & Sebe, N. (2019). *First Order Motion Model for Image Animation*. Conference on Neural Information Processing Systems (NeurIPS).

[5] Goodfellow, I., Bengio, Y., Courville, A. (2016). *Deep Learning*. MIT Press.

[6] Schwartz, O. (2018, November 12). *"You thought fake news was bad? Deep fakes are where truth goes to die"*. The Guardian.

[7] HOWE, D. C., & Posters, B. (2019). *Spectre Knows* (in "Alternate Realities", Site Gallery, Sheffield Doc/Fest, Sheffield, UK).

[8] Huang, S., Li, Q., Anil, C., Bao, X., Oore, S., & Grosse, R.B. (2019). *TimbreTron: A WaveNet(CycleGAN(CQT(Audio))) Pipeline for Musical Timbre Transfer*. ArXiv, abs/1811.09620.

[9] Kinsley, H. (2019, July 12). *Fake Voice Text to Speech Deep Learning ft. Elon Musk, Trump, Obama, and Joe Rogan* [Video]. YouTube. https://www.youtube.com/watch?v=6bFN2YkN6bo

[10] https://blog.thetobysiu.com/2020/04/04/extracting-audio-files/

[11] Brown, T.B., Mann, B.P., Ryder, N., … Amodei, D. (2020). *Language Models are Few-Shot Learners*.

[12] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A … Polosukhin, I. (2017). *Attention is All you Need*. ArXiv, abs/1706.03762.

[13] Bulat, A., & Tzimiropoulos, G. (2017). *How Far are We from Solving the 2D & 3D Face Alignment Problem?* (and a Dataset of 230,000 3D Facial Landmarks). 2017 IEEE International Conference on Computer Vision (ICCV), 1021-1030.

[14] https://github.com/alievk/avatarify

[15] Cudeiro, D., Bolkart, T., Laidlaw, C., Ranjan, A., & Black, M. (2019). *Capture, Learning, and Synthesis of 3D Speaking Styles*. Computer Vision and Pattern Recognition (CVPR), 10101-10111. Retrieved from http://voca.is.tue.mpg.de/

[16] https://blog.thetobysiu.com/2020/04/12/voca/

[17] PyTorch is an open source machine learning library based on the Torch library by Facebook.

[18] TensorFlow is an end-to-end open source machine learning platform by Google.

[19] Devlin, J., Chang, M., Lee, K., & Toutanova, K. (2019). *BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding*. ArXiv, abs/1810.04805.

[20] Yang, Z., Dai, Z., Yang, Y., Carbonell, J.G., Salakhutdinov, R., & Le, Q.V. (2019). *XLNet: Generalized Autoregressive Pretraining for Language Understanding*. NeurIPS.

[21] https://blog.thetobysiu.com/2020/04/15/witcher-books-data-extraction/

[22] https://github.com/minimaxir/gpt-2-simple

[23] Wolf, T., Debut, L., Sanh, V., Chaumond, J., Delangue, C., Moi, A., … Brew, J. (2019). *HuggingFace's Transformers: State-of-the-art Natural Language Processing*. ArXiv, abs/1910.03771.

[24] Ping, W., Peng, K., Gibiansky, A., Arik, S.Ö., Kannan, A., Narang, S., Raiman, J., & Miller, J.L. (2017). *Deep Voice 3: 2000-Speaker Neural Text-to-Speech*. ArXiv, abs/1710.07654.

[25] Shen, J., Pang, R., Weiss, R.J., Schuster, M., Jaitly, N., …Wu, Y. (2017). *Natural TTS Synthesis by Conditioning Wavenet on MEL Spectrogram Predictions*. 2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 4779-4783.

[26] Dauphin, Y., Fan, A., Auli, M., & Grangier, D. (2017). *Language Modeling with Gated Convolutional Networks*. ICML.

[27] https://github.com/tugstugi/pytorch-dc-tts

[28] https://blog.thetobysiu.com/2020/04/04/data-text-pre-processing/

[29] Prenger, R., Valle, R., & Catanzaro, B. (2018). *WaveGlow: A Flow-based Generative Network for Speech Synthesis*. CoRR.

[30] Keith, I. (2017). *The LJ Speech Dataset.* Retrieved from https://keithito.com/LJ-Speech-Dataset/

[31] https://github.com/DinoMan/speech-driven-animation

[32] https://github.com/mozilla/DeepSpeech

[33] Hannun, A.Y., Case, C., Casper, J., Catanzaro, B., Diamos, G., … Ng, A.Y. (2014). *Deep Speech: Scaling up end-to-end speech recognition*. ArXiv, abs/1412.5567.

[34] https://blog.thetobysiu.com/2020/05/08/deepstory-web-interface-and-flask/

[35] https://colab.research.google.com/github/alievk/avatarify/blob/master/avatarify.ipynb

[36] https://github.com/thetobysiu/audio-silence-normalize

[37] https://github.com/CSTR-Edinburgh/ophelia

[38] https://www.youtube.com/watch?v=bfGa6dMSnOo

[39] Mateas, M. Stern, A. (2003). *Façade: An Experiment in Building a Fully-Realized Interactive Drama*

[40] Brooker, C. (Writer), & Harris, O. (Director). (2013, February 11). Be Right Back. [Television series episode] In Brooker, C (Producer), *Black Mirror*. Channel 4.