The Office Character Dialogue Generation Using Images

Noah Foster (nfoster3), Chloe Griffin (dcgriffi), Abubakarr Jaye (ajaye), Christine Tseng (ctseng6)

Brown University, Deep Learning, Assistants (to the) Regional Programmer

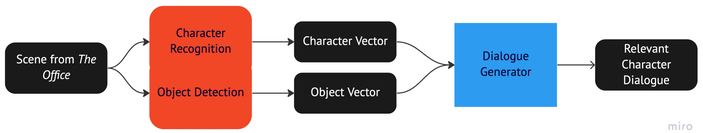
Introduction

Researchers have connected image and language models through image captioning and related projects. We expand on this concept by training a language model on an entire television show's script and using this model to"caption" a still frame with potential character dialogue.

Project Description

In this project, we build a dialogue generator for the TV show *The Office* (2005). In particular, the dialogue generator

1. takes in an image, or still frame, from the show,
2. identifies the present characters and objects in the scene, and
3. outputs lines based on what the characters would say in the given environment.



Methodology

We have two major network components in our model.

Image Model

- Character recognition ⟹
- Object detection ⟹

Language Model

- Dialogue relevant to characters
- Dialogue relevant to objects

Datasets

We use *The Office* Characters Kaggle dataset [7] to train the recognition model. It contains

- 6 characters: Angela, Dwight, Jim, Kevin, Michael, and Pam
- ≈ 250 images per character

We use *The Office* (US) - Complete Dialogue/Transcript Kaggle dataset [8] for the language model. It contains

- 6 categories: Season, Episode, Title, Speaker, and Line.
- ≈ 50,000 labeled lines

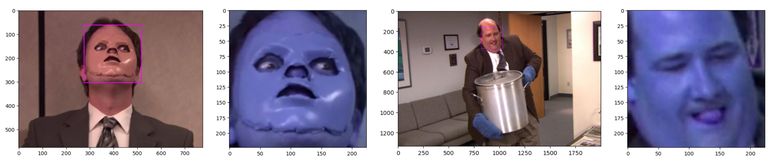


Figure 1. Examples of Preprocessed Images [3] [4]

Preprocessing

To prepare our images for the character recognition model as shown in Figure 1, we

- recognized faces with a Haar cascade classifier [13],
- cropped and resized the image, and
- converted to bgr color format with CV2.

To build character, object, and dialogue associations to fine-tune the language model, we

- organized the script by scene,
- matched characters with associated dialogue, and
- randomly selected nouns from the scenes.

Model Architecture

Our input image first passes through the **image model**, which uses

- The Office CNN Model, a custom transfer-learning model inspired by [14] which uses the 16-layer base model VGG16 trained on the vggFace2 dataset [9] containing 3.1 million facial images, and
- CLIP (Contrastive Language–Image Pre-training) [12] inspired by [11] to return the top 5 of 75 chosen nouns most likely detected in the image.

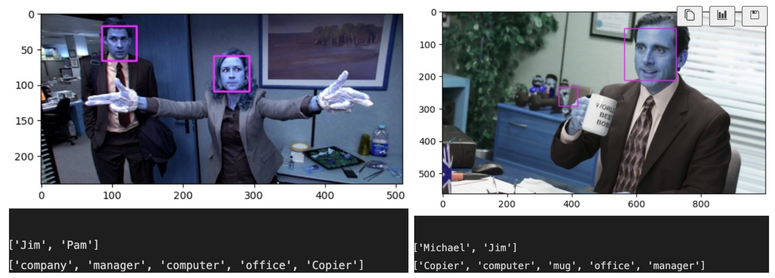


Figure 2. Examples of Image Model Output

The image model output (Fig. 2) is the input of the **language model**, which uses

- GPT2 [1] fine-tuned on *The Office* script dataset, inspired by [10], and preprocessed as previously described.

Results

Image Accuracy Metrics

Metric	The Office CNN
Training: Accuracy	0.9962
Loss: Categorical CE	0.0203
Testing: Accuracy	0.7428

LM Training Metrics

Epoch	Loss (SCCE)	Perplexity
5	2.4374	11.4432
10	2.1359	8.4646
15	1.9613	7.1085
20	1.8444	6.3243

Table 1. Training and Accuracy Metrics for the Two Network Components.

LM Validation Perplexities

GPT2 Tuned: 15.952	GPT2 Untuned: 30.888	GPT-J Untuned: 20.14

Example Outputs

Figure 3. Examples of Final Model Output

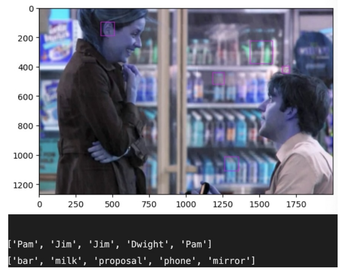
Discussion

Lessons Learned

- *Transfer Learning*: Our implementation of powerful models into our architecture was successful thanks to our research into transfer learning.
- *Identifying and Compiling Resources*: Understanding novel, high-performing models, and how they are compatible, requires time-consuming compilation of resources. Within the growing field, dealing with deprecated tools is also difficult to navigate.

Limitations

- *Object Extraction Incorporation*: Identified objects in still frames end up playing very heavily into the output dialogue. For example, a still frame of Pam and Jim's proposal would end up with dialogue about the milk detected in the back.



It is meaningful to modulate how present objects affect output dialogue.

- *Limited Data*: our image classification network was only trained for character classification on six main characters. This means we are only able to input still frames from the show that contain images of these six characters.

Future Work

- *Facial Expression Recognition*: given our preprocessing of still frames to only faces, facial expressions can be easily identified and thus represent the sentiments within a scene. This information can better inform the generation of dialogue that happens between characters.
- *Comparing More Language Model*: there are many large language models with powerful (e.g. zero-shot) capabilities; however, they are computationally costly. It would be interesting to see how results could differ from our current results.

References

[1] How to fine-tune GPT-2 for beginners | Kaggle.

[2] Image: Dwight and Jim, url = https://helios-i.mashable.com/imagery/articles/03O7JfLqPcz2jP9hdgZspjF/hero-image.fill.size$_1$200x675.v1632315010.png.

[3] Image: Dwight with mask on face, url = https://akns-images.eonline.com/eol$_i$$mages/Entire_site$/2020017/$rs_1$024$x$759 − 200117120856 − 1024 − The − $Office$ − $Stress$ − $Relief$.jpg.

[4] Image: Kevin Holding Chili, url = https://imagesvc.meredithcorp.io/v3/mm/image?url=https

[5] Image: Michael and others, url = https://www.hollywoodreporter.com/wp-content/uploads/2015/03/the$_o$$ffice_s$$eason_1c$$ast$.$jpg$.

[6] openai/clip-vit-base-patch32 · Hugging Face.

[7] The Office Characters | Kaggle.

[8] The Office (US) - Complete Dialogue/Transcript | Kaggle.

[9] Qiong Cao, Li Shen, Weidi Xie, Omkar Parkhi, and Andrew Zisserman. VGGFace2 Dataset.

[10] Murat Karakaya. Training a Hugging Face causal language model from scratch (TensorFlow).ipynb - Colaboratory, 7 2022.

[11] Open AI. Interacting with CLIP.

[12] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning Transferable Visual Models From Natural Language Supervision.

[13] Paul Viola and Michael Jones. Rapid Object Detection using a Boosted Cascade of Simple Features. 2001.

[14] Wei-Meng Lee. Implementing Face Recognition Using Deep Learning and Support Vector Machines, 8 2022.

This project was completed for the Fall 2022 Deep Learning Day in the course CS2470