Analytical Film Analysis

Arden Rasmussen

This project is utilizing computer science and machine learning to construct an algorithm to analytically compare different films, and to compute a numerical value that represents the similarity between the films. In general terms, the program that I have made considers a set of films, and compares features that appear in each films, and attempts to numerically represent the similarity between the two films. After doing this process for each pair of films individually, then the weights of the connections between films are used to construct a graph, were the length of the connection between the films indicates a greater similarity between them.

All of the code that I have written has been in python, and using the OpenCV library. The exact process that my program executes is to first load the video, and resizes the video file to be 100×100. Then the program extracts 1000 frames from the video. Then for each frame a process of feature extraction is done. This process is relatively complex, but the result is a list of values that represents the content of the frame, this process is called the vectorization of the images. The vectorization process is the conversion from a two dimensional image with a lot of content, to a one dimensional vector of values which represents the same content of the image. Once each image has been vectorized, then the vectors of each film are compared through the use of the cos function, and compares the angles between the vectors. Vectors with no similarity will be perpendicular, and the result of this function will be 0, and vectors that are exactly the same, indicated identical frames, will result in a very large number. By computing the similarity of each of the selected frames from one film with each of the frames from the other film, could result in some erroneous values, so we additionally scale the similarity of vectors in relation to their temporal proximity. That is to say that similar frames that occurs at similar points in time are more similar than similar frames at opposite ends of the film. An example of this could be credits, opening credits and end credits look very similar, but are very different, and so they should not be considered similar. Computing the average of the weighted similarity of all the frames/vectors, we can consider this to be a rough approximation of the similarity between the films.

The next step of the process is to generate the graph, this is primarily handled by the D3 library.

I think that this project was interesting, because we have discussed at significant length which films are similar, and whey they are similar. So having implemented a method for a computer to consider the similarity between the films is interesting, because it is not identical to how we may consider the similarities. This algorithm is commonly used in most image search algorithms, and so being able to understand how computer vision differs from that of human vision I find quite interesting.