Text-to-video dataset creation from videos with no sound

Problem:

Text-to-video generation models recently reached a very high-qulity due to the rapid development in diffusion model. Most of the current text-to-video models use three separate models to perform generation. And while this approach works well for generation of short videos it may become a problem for longer video generation as the longer the video the stronger logical connection between its parts should be

Usage

The following method can help to create a dataset for each part of the generation model and fine-tune it with the exactly similar data in the latent space. As we know current approach to Large Language Models is to traing a big models with several billions parameters and after that fine-tune it on a more niche dataset. This method approach the same for a video generation model dataset which aims to create a long logical videos.

Solution

Video dataset → multiple chapters

How to divide long video to chunks? Extract frames in certain periods and measure with some metric. As in future we will operate in vector space probably we should measure vector similaruty further and now don’t want to use pixel-based because it can be too depended on light. We left with feature-based( edges, corners) to compare similarity. Here we also want to measure distance between our features as it will mean that our object is the same in every frame and we have still the same scene (chapter). After we found different images should measure images nearby to found chunk borders.

Metrics to use: SSIM, FSIM.

Algorythm:

For frame\_n in video:

if SSIM(frame\_n, frame\_n+1) < threshhold:

chapter\_1.append(chapter\_start, frame\_n)

chapter\_start = frame\_n+1

A chapter → multiple single frames that can be unified into short video

I think here we should convert our images to vectors and measure cosine similarity as from now we need a consistency between images and text and we can achieve that when operating with both images and text in the latent subspace.

Metrics to use: cosine similarity

Algotythm:

For frame\_n in chapter:

if cosine\_similarity(frame\_n, frame\_n+1) < threshhold,

Frame\_list.append(frame\_n, frame\_n+1)

Output: frame\_list, consist of n different frames which are different in CLIP latent space

A multiple frames → long descriptions

We extract information from video(frames) not from audio (it is not that hard with speech-to-text and summarization. Therefore, we don’t use audio and operate with frames only. I think audio and video should be treated as a separate channels and should have only timiline in common. As we already have separate frames we now use image-to-text with to get a description but we need our description to be logical so we should use a conditional text generation. We start from our first image and then generate further description with a condition of second image description

Algorythm:

For frame in frame\_list:

generate\_text|previous\_text,frame

A long description → short summary

LLM summarization, probably should take fine-tuned LLM, for metrics we can use BLEU, Rouge, Meteor, BERTScore.

Alternatives

For videos with speech subtitles summarization can be used. First is speech-to-text model and after that text summarization

Claims