Progress Report

Justin Shin

George Mason University
April 11, 2023

1. Introduction

Sponsorblock is a browser extension that automatically skips Youtube sponsor ads during video playback. It is a crowd-sourced effort to accurately label sponsor ad segments to allow the extensionto skip over those segments. Users can vote on the accuracy of the sponsor segment on whether they were placed appropriately or not in which case, with sufficient downvotes, the segment becomes deactivated. This process of labelling the segments have been done by human volunteers since the project's inception, but with ready access to the SponsorBlock database through the free API, this process can be automated using Deep Learning Neural Networks. The goal of this project is to build a model that can predict those sponsor segments as accurately as a human volunteer.

2. Related Work

There have already been a few attempts at automating the sponsor ad labelling process but none are in active development. The following projects were the few that made notable attempts. All projects have leveraged the SponsorBlock API for ground truth sponsor segment labels.

2.1. NeuralBlock

NeuralBlock used a bidirectional-LSTM RNN that trained on the transcripts downloaded from YouTube. It tokenized the top 10,000 words found in the sponsor segments to train the model. At one point, the creator of SponsorBlock considered this model to automate the labelling process but was quickly abandoned when the creator halted development.

2.2. reBlock

reBlock used a DeBERTa [1] model to predict sponsor segments. It was trained on over 31,000 video transcripts. It was the result of a hackathon project that won first place.

2.3. DeepSponsorBlock

DeepSponsorBlock was a Stanford C230 project that used the video frames to predict the sponsor segments instead of using the transcript. The advantage to this method is that it can predict sponsor segments regardless of language as it is not dependent on natural language nor audio. The architecture consists of an CNN that extracts features from the video frames and an RNN that predicts the start and end timestamps for the sponsor ad. One limitation is that the dataset only consisted of videos that had exactly one sponsor ad, where the average YouTube video may have two or more sponsor ads. The resulting model suffered from overfitting with an accuracy of 79%.

3. Method

3.1. Dataset

The dataset consists of sentences with a value denoting whether the sentence was from a sponsor ad or not with either a 1 or a 0 respectively. The transcripts were downloaded using the Python package called youtube-dl and the sponsor segments were identified with the SponsorBlock API. Only a small number of transcripts were allowed to be optained due to the rate limitations from YouTube. The current dataset consists of 100 video transcripts which consists of a total of 6000 sentences. Even though the YouTube video may have provided a handwritten transcript from the YouTuber, the auto-generated transcripts were used instead since the majority of the videos only supported auto-generated transcripts. This is to keep the dataset consistent as auto-generated transcripts do not have any punctuation.

Most of the project was spent on creating a workable dataset. One challenge was to avoid alarming YouTube servers from downloading too many transcripts. Even though only 100 video transcripts were downloaded, some were downloaded multiple times to test the package youtube-dl as well as to train the model on auto-generated transcripts and handwritten transcripts. The most challenging problem was from parsing the auto-generated transcripts. The handwritten transcripts provided accurate timestamps denoting the accurate start and end times for each sentence due to the nature of the transcripts being manually typed in. The auto-generated transcripts had multiple timestamps for each sentence since the transcript generator could not neatly separate the sentences. Much time was spent isolating each sentence and pairing with a single accurate timestamp.

3.2. Model

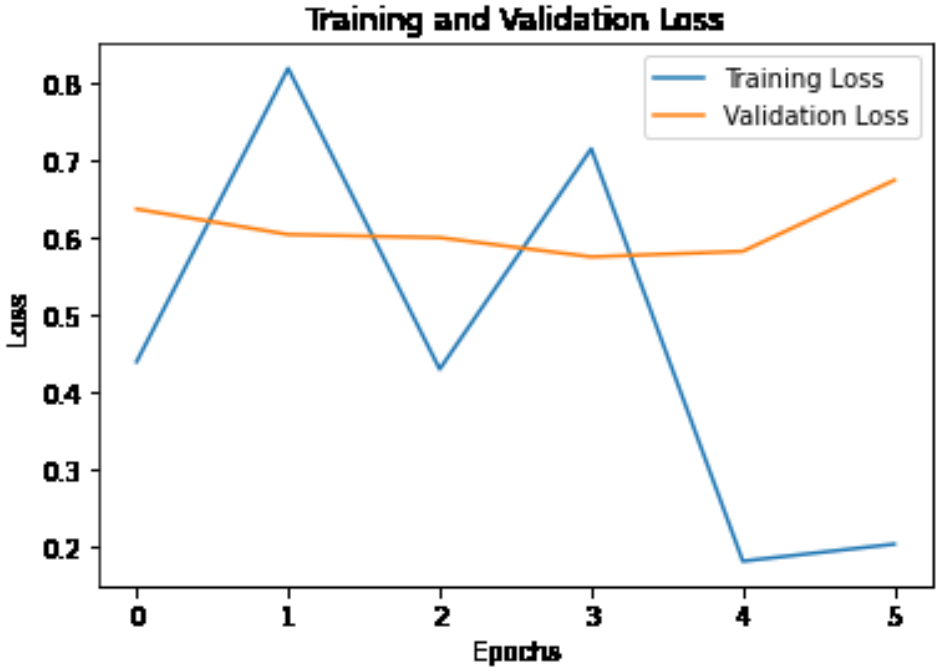
A pretrained RoBERTa [2] model was used for this project, specifically, the roberta-base model from Hugging Face was used. The BERT [3] model was originally proposed to be used but the Roberta was used instead because it was more performant especially in terms of memory usage. Specifically the bert-base-uncased model from Hugging Face was tested. The current model is a binary text classification model that takes each sentence which has either a value of 1, if it was from a sponsor ad, and 0, otherwise. The model takes a sentence as input as outputs a corresponding value of 1 or 0. The learning rate was set to 4e-5 and the AdamW optimizer was used and was trained on a single Nvidia T4 GPU.

4. Prelimary Results

Table 1. Results

MCC	AUROC	Loss
0.459	0.794	0.616

The model outputted acceptable results. It was only trained on 5 epochs because the model began to degrade and overfit with higher number of epochs. The Matthews Correlation Coefficient (MCC) is a performance score that measures the outputs of the confusion matrix. With a score of 0.459, the model performed modestly well. The AUROC was a passable 0.794 where 1 is a perfect score and 0.5 is the result of randomly guessing. Finally, the evaluation loss was 0.616.



Although the model performs passably well, the model currently still suffers from misclassification. Sometimes the model performs better than expected. It is even able to classify segments where the YouTuber is self-promoting (not to be confused with sponsor ads). But many times it is classifying non-sponsor segments incorrectly resulting in a noisy output where a non-sponsor segment should consist of only 0 values but instead have 1's sprinkled throughout.

5. Conclusion and Future Work

The model performed with passable results but could further be improved on throughout the rest of the semester. There were many things learned during the research and building of the model, such as the limitations of the current model its possible solutions.

The current model is a binary text classification model that classifies a given sentence as either part of a sponsor ad or not. While the current model seems to perform modestly at this task, it fails to understand the context of the entire transcript as a whole. A more appropriate model is a text segmentation model that can take an entire transcript and split it into sponsor or non-sponsor segments. An even more ambitious improvement would be to segment the transcript into more than two categories.

The SponsorBlock extension not only supports timestamps for sponsor segments but also other categories such as: "Highlight" that shows clips of the video at the beginning of the video, "Interaction Reminder" where the YouTuber performs a call-to-action by telling its audience to "subscribe" to the channel or "like" the video, and "Self Promotion" where the YouTuber will advertise for their own product. At first glance, it may seem too ambitious but as long as the original script can accurately segment the transcript using SponsorBlock API, it shold not take much more complications to extend the binary class labelling to multi-class labelling.

Only two pretrained models have been experimented with but other such transformer models should be trained with as well. It may also be worthwhile to compare with other types of models such as RNN. Two of the three related work projects mentioned above use Bidirectional LSTM's for performant classification. It may be more advantageous to use an LSTM model since it works better with smaller datasets than transformers.

Though much of the suggested improvements and changes make it seem as though the project is pivoting away from the originally proposed goal, the original goal is still unchanged; to creating a model that can lessen the burden of the human SponsorBlock volunteers by automatically timestamping sponsor segments in YouTube videos.

References

1. Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen, "DeBERTa: Decoding-enhanced BERT with Disentangled Attention}„" *CoRR* (2017).
2. Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov, "RoBERTa: A Robustly Optimized BERT Pretraining Approach," *CoRR* (2019).
3. Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova, *BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding* (2018).