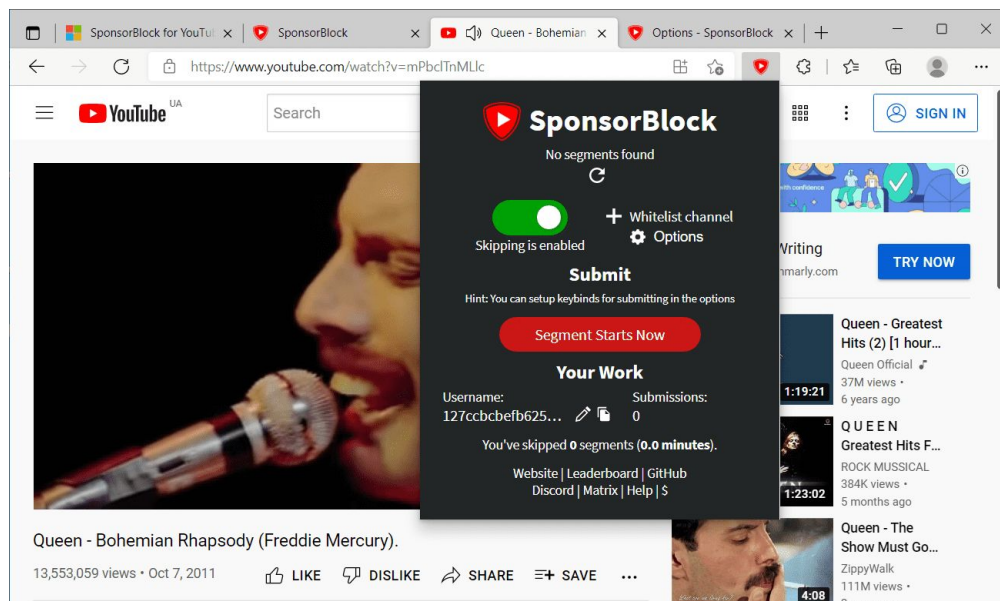


# DeepLearningYouTubeSponsorBlocker

Justin Shin

# What is SponsorBlock?

- Browser extension
- Automatically skips sponsor ads (and more)
- Crowdsourced data
  - Voting system



# My YouTube



# Motivation

- Zero contribution to majority of videos
  - Need to wait for new videos
  - Sometimes only skips some ads
- 
- Solution: Deep Learning model that can automate this

# Related Work

## 1. NeuralBlock

- <https://github.com/andrewzlee/NeuralBlock>

## 2. DeepSponsorBlock

- <https://github.com/DeepSponsorBlock/DeepSponsorBlock>

## 3. reBlock

- <https://github.com/MonliH/reBlock>

## 4. SponsorBlock-ML

- <https://github.com/xenova/sponsorblock-ml>

# Related Work: NeuralBlock

- First serious model to attempt this problem
- Endorsed by SponsorBlock creator
- BiLSTM: predict start and end of segment

## NeuralBlock: An In-Video YouTube Sponsor Detector

### Welcome!

This is the project page for NeuralBlock (NB), a neural network built using Keras/Tensorflow that detects in-video YouTube sponsorships. You can find the code for the project on [Github](#). Additionally, this project provides automoderator support to Ajay's [SponsorBlock](#), whom supplied the crowd-sourced sponsor labels needed to train NB.

### Demo

Submit a video ID and see where the sponsorships are. For true predictions, submit videos published after March 3, 2020. Videos without English captions currently cannot be processed.

Sample: <https://www.youtube.com/watch?v=QqEuO5im1nA>

### Examples

Half As Interesting	QqEuO5im1nA
Linus Tech Tips	B821HqH-dWI
Domo Wilson	V1YMJGuYPnA
Mentour Pilot	g1rpVMKz2S0
MrBeast (No Sponsorship)	HBMmK1c44sE

Video ID:

Submit

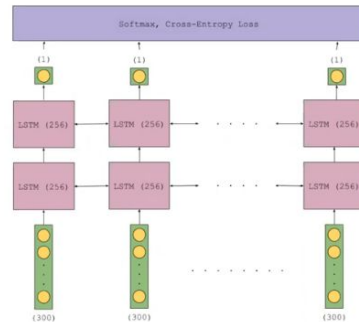
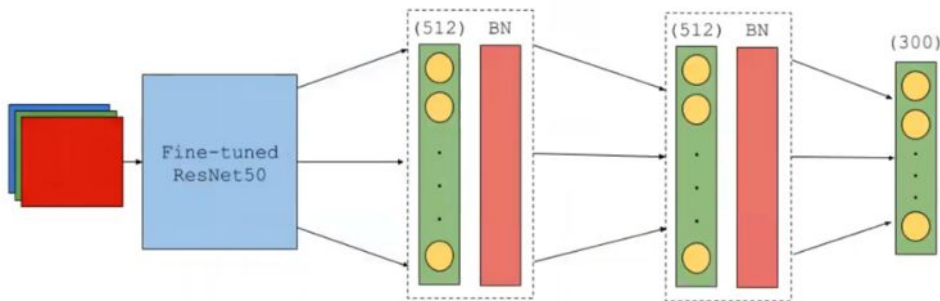
This project was created by [Andrew Lee](#).

If you have any questions, suggestions, or comments email me at [leezandrew@gmail.com](mailto:leezandrew@gmail.com).

Special thanks to [Joe Dowd](#) for his help in setting up the backend.

# Related Work: DeepSponsorBlock

- Stanford CS230 project
- Uses video frames instead of transcripts
- Architecture:
  - Resnet50 CNN: feature extraction from frame
  - BiLSTM: predict start and end of segment
- Suffered from overfitting



# Related Work: reBlock

- Hackathon project
- DeBERTa: token classification
  - Max: 4096 tokens in one pass
  - ["this", "video", "is", "sponsored", "by", "NordVPN", "so", ...]  
[1, 1, 1, 1, 1, 1, 0, ...]

reBlock

Block youtube sponsors using state-of-the-art natural language models! [See Code on Github](#)

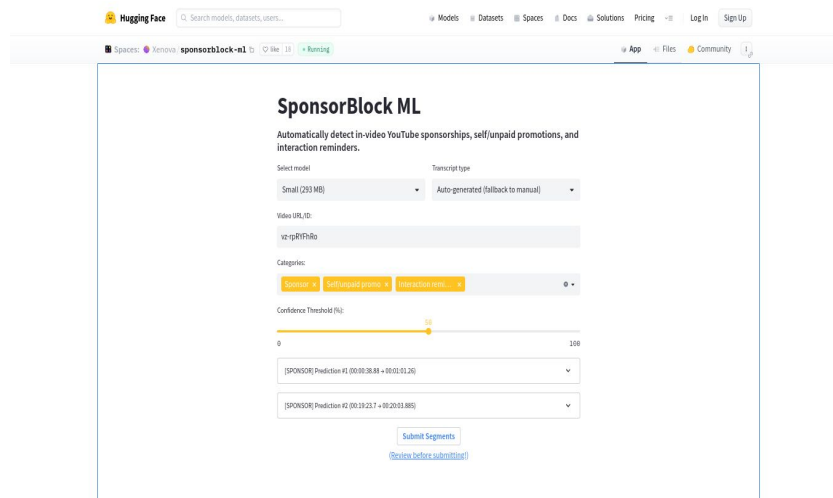
Enter Youtube URL:

[View Without Sponsors](#) →



# Related Work: SponsorBlock-ML

- Currently used to check submissions
- Architecture:
  - t5: Summarizer
  - bert: Classifier
- Can detect:
  - Sponsor segments
  - Interactions
  - Self Promotions



The screenshot shows the Hugging Face interface for the 'SponsorBlock-ML' model. The page title is 'SponsorBlock ML' with the description 'Automatically detect in-video YouTube sponsorships, self/unpaid promotions, and interaction reminders.' The interface includes a 'Select model' dropdown set to 'Small (293 MB)' and a 'Transcript type' dropdown set to 'Auto-generated (fallback to manual)'. There is a 'Video URL/ID' input field containing 'yz-rpFTHRo'. Below this is a 'Categories' section with three buttons: 'Sponsor', 'Self/unpaid promo', and 'How to interact', with a dropdown arrow. A 'Confidence Threshold (%)' slider is set to 50, with a range from 0 to 100. Two text boxes show predictions: '[SPONSOR] Prediction #1: (00:00:38.88 - 00:01:01.26)' and '[SPONSOR] Prediction #2: (00:10:23.7 - 00:20:03.88)'. At the bottom, there is a 'Submit Segments' button and a link '(Review before submitting!)'.

# Dataset

- SponsorBlock API (14M+ contributions)
  - 10K most votes
- (Unofficial) Youtube Transcript API
  - Auto-generated captions only
  - <https://github.com/jdepoix/youtube-transcript-api>

# Model: v1

- Dataset:
  - <label>, <caption>
  - 1154 transcripts
- RoBERTa: classification of each line
- Parameters:
  - Learning rate:  $4e-5$
  - Optimizer: AdamW
  - Epochs: 5

# Metric: Matthew Correlation Coefficient (MCC)

- Precision and Recall score using TP, TN, FP, FN
- Quality of classification:
  - +1: perfect prediction
  - 0: random prediction
  - -1: inverse prediction

	$y = 1$	$y = 0$	total
$x = 1$	$n_{11}$	$n_{10}$	$n_{1\bullet}$
$x = 0$	$n_{01}$	$n_{00}$	$n_{0\bullet}$
total	$n_{\bullet 1}$	$n_{\bullet 0}$	$n$

$$\phi = \frac{n_{11}n_{00} - n_{10}n_{01}}{\sqrt{n_{1\bullet}n_{0\bullet}n_{\bullet 0}n_{\bullet 1}}}.$$

# Results: v1

<b>MCC</b>	0.4625
<b>TP</b>	434
<b>TN</b>	410
<b>FP</b>	156
<b>FN</b>	154

# Model: v2 - Data

- Segmentation Dataset:
  - <label>, <caption1> <caption2>
  - Size: 2M+
  - Undersampled -> 4393
- Multiclass Classification Dataset:
  - <label>, <segment>
  - 0: None, 1: Sponsor, 2: Self-promo, 3: Interaction
  - Size: 10992

# Model: v2 - models

- Topic Segmentation
  - BERT, RoBERTa, XLNet, XLM-RoBERTa
- Text Classification
  - RoBERTa
- Parameters:
  - Learning rate:  $4e-5$
  - Optimizer: AdamW
  - Epochs: 5

# Results: v2

## Segmentation

	BERT	RoBERTa	XLNET	XLM_RoBERTa
MCC	0.5826	0.6131	0.5472	0.5947
TP	1720	1750	1535	1703
TN	1755	1792	1847	1797
FP	423	386	331	381
FN	495	465	680	512

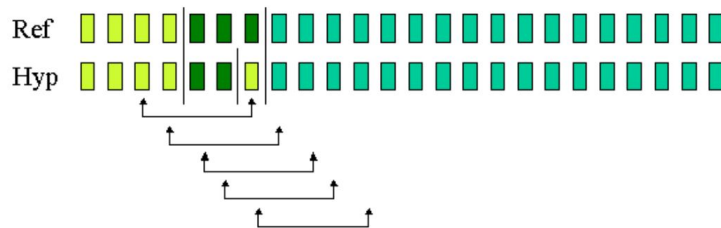
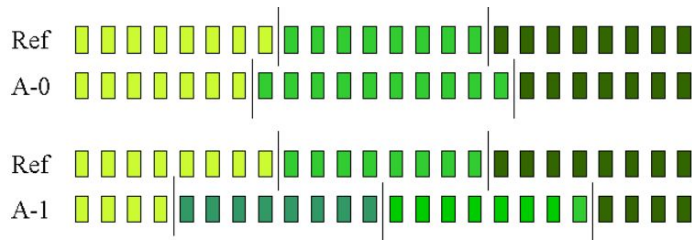
## Classification

MCC	0.8785
-----	--------



# Metric: Problem

- MCC: penalizes near-misses
- Solution: Pk score (lower the better)



# Conclusion and Future Work

- Pk metric
- Sentence-BERT
- Use Whisper if no transcript available
- Integrate video frame data

Thank You