NLA PROJECT OUTLINE - *Detecting Child Unsafe Videos using Transcripts*

Introduction

According to the statistics released by YouTube, the user base of YouTube is over a billion users and 400 hrs of viewable content is getting uploaded in 60 seconds. YouTube provides ease of publishing and is highly leveraged by content producers to host their video content on YouTube.

Our work mainly deals with detecting unsafe content on Youtube from the transcripts of videos. We focuses on cartoon videos which are typically watched by children. Various cartoon production houses have uploaded their trademark cartoon series on YouTube for wider publicity and brand building. In lieu of our focus on content being watched by kids, we restricted ourselves to collection of cartoon videos only.

Two types of child unsafe content are studied, one, which is sexually explicit and second, which contains violence. By violence we include Abusive Speech too, as our main analysis focuses on text. Put together, and we have three unsafe classes namely sexual, violent and both.

Data Collection

We plan on collecting transcripts from cartoon videos and shonen anime which are mostly targeted for kids under 15 years. Youtube-dl is used to get the transcripts from the youtube video links.

After analyzing statistical information about video consumption trends amongst the young, we have identified the following channels of communication to have the highest possibility of unsafe content, cross-referenced by the extent of consumption of said platforms by individuals:

- General interest portals (*9Gag, Reddit, etc*)
- Social networking (*Instagram, Facebook, etc)*
- Email
- Video content distributors (*YouTube, Netflix, etc)*

- Kids entertainment websites
- Online games *(Twitch live streams, YouTube, etc)*

Methodology

We plan on using sequence model architectures such as LSTM and BLSTM with character grams as inputs for training. BLSTM and LSTM models usually consists of three sequential layers—(a) Embedding Layer (b) Bidirectional LSTM (BLSTM) Layer or LSTM Layer and (c) Fully Connected (FC) layer.

1. To deal with complex structured conversation data, we plan to use character level trigrams to represent each conversation. Character n grams are for words out of the vocabulary. We also use word embeddings like glove fastext etc.
2. Sequence of character trigrams sent through an embedding layer which takes one-hot vector of each character trigram in the sentence and learns the lower dimension representation for it.
3. Bi-Directional LSTMs have the ability to capture much richer sequential patterns from both directions of a sequence.
4. Each character trigram from the previous embedding layer is fed as a sequence of trigrams, to the BLSTM layer.
5. The output of the BLSTM layer is given as input to a Fully Connected (FC) layer which models the interactions between these features.
6. The final softmax node in the FC layer outputs the probability of the sentence belonging to the inappropriate class.

This choice of model was made after trying out various architectures and techniques (*both classical and deep*) to classify base transcript sentences. The results from the models, viz. Deep learning *bi-LSTM*, ensemble voting approach, and classical POS and NER tagging have been included as part of the midway section. After comparing the accuracies, F1-scores and overall performance, the bi-*LSTM* model was chosen to be the core of our pipeline. We are looking into voting-based systems and given the relatively

high accuracies of the bi-LSTM and classical model, a combination of the two might lead to slightly better results.

Timeline & Deliverables

There will be two deliverables as part of this project

- A midway submission consisting of the initial model and detector at a sentence level.
- The final submission that will contain the end-to-end pipeline with all the relevant scripts and data required for replicating the results.

Date	**Goal**
~~20th February~~	Project Outline Report - **DONE**
~~1st March~~	Deliverables outline - **DONE**
~~22nd March~~	Interim report - **DONE**
16th April	Final presentation

Midway Checkpoint - *Model analysis and choice*

Timeline	**Tasks**
Week 1	Data analysis and collection
Week 2	Literature review and model aggregation
Week 3	Exams
Week 4	Model training and final choice

Data Analysis and Collection

After looking at average statistics from a number of websites regarding the kind of content teens and young children consume, we were able to prepare a wholistic dataset covering all sources mentioned prior. There are around 30 videos spanning across all genres for an average of 7 videos per genre. These videos are approximately 5-7 minutes each.

Once the links were collected were collected, we ran the a script that collected and labelled each line of each transcript with the following tags:

- YouTube ID: The unique uRL of the video from which the text is obtained.
- Start time: With reference to the start of the given video
- End time: The end of the text in the video
- Text: The actual text contained/spoken within the start/end

Now, because we are dealing with the detection of unsafe content for children, our labels needn't be very detailed. It is for this reason, the data is labeled either hate speech (**0**), offensive language **(1)**, or neither (**2**). In the future, we would like a finer level of detection so as to give a more in-depth analysis of a given test video.

For the model choice, to prevent data-based bias and skew in the accuracies, we have chosen a common dataset to train and validate the models on. All the contenders for the base model in our pipeline have been trained on the Davidson hate speech dataset *(Automated Hate Speech Detection **ICWSM***).

Model Choice and Analysis

The three models we tested on the standard dataset were the bi-LSTM model, classical NER/POS based model, and an ensemble bayes + LSTM model. **The precision, recall and F1-scores of the classical approach and the bi-LSTM model were 91 and 94% respectively**.

While the architecture of the voting based method is quite good, the use of *less* powerful models (*naive bayes, LSTM*) has pushed us to create a voting model using a combination of the bi-LSTM and POS-based models. This *may* lead to better performance, which will help the later part of the pipeline. We are currently investigating into this and might shift to such an approach, given the time constraints.

Approach 1: (Classical Approach)

This model has been proposed by "Automated Hate Speech Detection and the Problem of Offensive Language." ICWSM.

Features

- We lowercased each sentence and stemmed it using the Porter stemmer, then create bigram, unigram, and trigram features, each weighted by its TF-IDF.
- To capture information about the syntactic structure we use NLTK (Bird, Loper, and Klein 2009) to construct Penn Part-of-Speech (POS) tag unigrams, bigrams, and trigrams.
- To capture the quality of each tweet we use modified Flesch-Kincaid Grade Level and Flesch Reading Ease scores, where the number of sentences is fixed at one.
- We also use a sentiment lexicon designed for social media to assign sentiment scores to each tweet (Hutto and Gilbert 2014).

- We also include binary and count indicators for features for the number of characters, words, and syllables in each sentence.

Model:

- We first use a logistic regression with L1 regularization to reduce the dimensionality of the data.
- We tested each model using 5-fold cross validation, holding out 10% of the sample for evaluation to help prevent overfitting.
- After using a grid-search to iterate over the models and parameters we find that the Logistic Regression and Linear SVM tended to perform significantly better than other models.
- We decided to use a logistic regression with L2 regularization for the final model.
- We trained the final model using the Davidson dataset and used it to predict the label for each sentence of Youtube Transcript Dataset.
- We use a one-versus-rest framework where a separate classifier is trained for each class and the class label with the highest predicted probability across all classifiers is assigned to each tweet.
- All modeling was performing using scikit-learn.

Results

	precision	recall	f1-score	support
0	0.47	0.56	0.51	1430
1	0.97	0.92	0.94	19190
2	0.83	0.96	0.89	4163
micro avg	0.91	0.91	0.91	24783
macro avg	0.75	0.81	0.78	24783
weighted avg	0.91	0.91	0.91	24783

Approach 2: (bi-LSTM)

We used sequence model architectures i.e., BLSTM with character grams as inputs for training. BLSTM and LSTM models usually consists of three sequential layers—(a) Embedding Layer (b) Bidirectional LSTM (BLSTM) Layer or LSTM Layer and (c) Fully Connected (FC) layer.

Features

- To deal with complex structured conversation data, we plan to use word level trigrams to represent each conversation.
- Sequence of trigrams sent through an embedding layer which takes one-hot vector of each character trigram in the sentence and learns the lower dimension representation for it.

Model

1. Bi-Directional LSTMs have the ability to capture much richer sequential patterns from both directions of a sequence.
2. Each character trigram from the previous embedding layer is fed as a sequence of trigrams, to the BLSTM layer.
3. The output of the BLSTM layer is given as input to a Fully Connected (FC) layer which models the interactions between these features.
4. The final softmax node in the FC layer outputs the probability of the sentence belonging to the inappropriate class.

Results

```
Epoch 21/30
19783/19783 [==============================] - 86s 4ms/step - loss: -0.8691 - acc: 0.9391 - val_loss: 0.1019 - val_ac
c: 0.8428
Epoch 22/30
19783/19783 [==============================] - 80s 4ms/step - loss: -0.8713 - acc: 0.9400 - val_loss: 0.1204 - val_ac
c: 0.8408
Epoch 23/30
19783/19783 [==============================] - 81s 4ms/step - loss: -0.8731 - acc: 0.9404 - val_loss: 0.2340 - val_ac
c: 0.8402
Epoch 24/30
19783/19783 [==============================] - 74s 4ms/step - loss: -0.8730 - acc: 0.9403 - val_loss: 0.2093 - val_ac
c: 0.8390
Epoch 25/30
19783/19783 [==============================] - 78s 4ms/step - loss: -0.8710 - acc: 0.9394 - val_loss: 0.2144 - val_ac
c: 0.8392
Epoch 26/30
19783/19783 [==============================] - 76s 4ms/step - loss: -0.8726 - acc: 0.9400 - val_loss: 0.1851 - val_ac
c: 0.8390
Epoch 27/30
19783/19783 [==============================] - 85s 4ms/step - loss: -0.8750 - acc: 0.9404 - val_loss: 0.1948 - val_ac
c: 0.8386
Epoch 28/30
19783/19783 [==============================] - 78s 4ms/step - loss: -0.8737 - acc: 0.9403 - val_loss: 0.2013 - val_ac
c: 0.8402
Epoch 29/30
19783/19783 [==============================] - 79s 4ms/step - loss: -0.8745 - acc: 0.9403 - val_loss: 0.2292 - val_ac
c: 0.8344
Epoch 30/30
19783/19783 [==============================] - 91s 5ms/step - loss: -0.8741 - acc: 0.9400 - val_loss: 0.2117 - val_ac
c: 0.8352
```

Approach 3 (Ensemble Learning)

Ensemble learning through weighted voting systems, which include precision score, CEN score and equal voting. Councilor classifiers are:

- Voting Model
- LSTM Model
- Bayes Model

After evaluating the performance of the various voting metrics, viz. weighted average with precision and CEN scores vs. equal voting, the voting method that will be used in the final combination is the former. This, with the classification methods of the prior two approached should give us the best performance. Additionally, information from the classical method

can be used in the later clustering and analysis step. Since we're planning to continue this further, a later step is to provide a finer analysis of the type of content and possibly give an appropriate age rating, though this might require some additional work.

Analysis

- We take each input video and identify the timestamp where child unsafe content occurs.
- We take all such instances where unsafe content occurs in a video.
- We then mark all occurrences and generate a heatmap of the video timeline. Depending on how the content is structured/detected, we then assign an overall score and label to the video.

Conclusion

At the end of the project, hopefully we will be able to rate videos from youtube as child unsafe from a scale of 0-10 using text from transcripts of videos as the only input.

Team Members

- D. Shritishma Reddy (20161165)
- Pratik Kamble (20161135)

Project Details

- Title: Detecting child unsafe videos on YouTube using Transcripts
- Project #14.2