Predicting Video Memorability

Comparing multiple machine learning models

Gaurav Negi
StudentID: 19210869
Gaurav.negi2@mail.dcu.ie

Abstract— **In this paper I will be exploring the results of various machine learning models applied on the dataset provided for the MediaEval 2018 "Video Memorability" challenge. The challenge requires us to predict short-term and long-term memorability score. After exploring multiple models, we got best results from a tuned neural network.**

Keywords— HMP Features, C3D Features, Semantic Features (TF-IDF).

*Introduction (*Heading 1*)*

Predicting video memorability challenge focusses on predicting the memorability score for any given video. This problem originated as the media platforms such as Facebook, Netflix, google search engine deal with big data daily and need for new techniques that can help organize and retrieve the digital content[1]. The challenge is divided into two sub tasks:

i. Short-term Memorability: the task involves predicting a 'short-term' memorability score.
ii. Long-term Memorability: the task involves predicting a 'long-term' memorability score.

Data is provided by DCU, which includes 6000 videos in dev-set and 2000 videos in test-set. Both sets have annotations and titles (captions) associated with the videos. The ground-truth is available only for the dev set. Given videos are short and soundless. To facilitate participations we are also provided with pre-computed features such as HMP features (the histogram of motion patterns in each video) and C3D spatio-temporal visual features that are obtained by extracting the output of the final classification layer of the C3D model, a 3-dimensional convolutional network proposed for generic video analysis.[2] To measure the score of submissions on ground truth of test data spearman's correlation coefficient is used. I compared multiple models with different parameters to get the highest spearman correlation coefficient and the used that model to get our final predictions.

Based on the previous work done on this challenge, I will be focusing on following feature combinations:

i. Semantic Captions
ii. Semantic Captions with weights
iii. Captions and HMP features
iv. Captions and C3D features

I. Literature Review

In [3], The research team used a 2 layer CNN model, with each model consisting of a 2D convolution, batch normalization, relu activation and a max pool operation. In the end they all are vectorized and a fully connected linear is applied. In [4], the team used different models on different features. First, they divide features into low and high level, depending on the type of information they contain. They SVR on low level features and used CNN-LSTM and Bi-LSTM on high level features, features contain more information. In [5], the research team used three model on different video and sematic features. And realise that caption with weights are generating better result as compared to rest of the model. They used three model, LASSO, LaMem and Linear Regression.

II. Approach

This section consists of approach I used to produce good results.

A. Models

I initially started with Linear Regression and decision trees but then alter removed these models as their performance was very poor compared to the ones mentioned below.

I used following main models for all features:

i. Random Forest Regression
ii. Support Vector Regressor
iii. Sequential Neural Network

These models were used with different parameters and parameter tuning was also performed.

B. Features and data-processing

- **Semantic feature** – When Captions were used, the models performed very well as compared to using Video Features alone. Using Corpus of stopword, downloaded using NLTK library, we removed stopword from the captions. Also did some cleaning in the captions. Then each caption, corresponds to a video, transformed into TF-IDF score, and supplied to the models in form of vector. I also explored another method where I assigned specific weights to certain words such as 'woman: 16 ', 'eating: 15' etc[6]. Then spearman 's correlation coefficient was used for evaluation. Captions with weight got slightly better results in random forest but surprisingly the captions without external weights got best results using neural network.

Video Features with Semantic features – In this Model a combination of Video and Caption features were used. The HMP and C3D features both performed poor results. Since captions performed good, I combined the semantic features with video features which did improve the results, but the changes were not significant enough and captions still outperformed the performance.

C. *Best Model Tuning:*

Out of all the models Neural Network with captions performed best. To create a baseline model for this I used previously explored parameters obtained using GridSearch. The best parameters for captions were:

i. Droprate = 0.3 and weight constraint: 3

ii. Neuron Activation function = "selu"

iii. Learning rate optimizer = "adadelta"

iv. Epochs = 50

In order to tweak further I used classical methods of adapting learning rate for stochastic gradient descent optimization. This approach has the benefit of making large changes at the beginning of the training procedures when large values are used and decreasing the learning rate such that smaller rate and smaller training updates are made to weights later in training procedure. This has the ability of quickly learning good weights early and fine tuning them later. I decided to use two learning schedule approaches[7]:

Time-based learning Rate Schedule: In this we decrease the learning rate gradually based on the epoch. We will use the built-in time-based schedule of Keras. The decay argument of SGG optimizer is used in this schedule and follows below equation:

$$\text{LearningRate} = \text{LearningRate} * 1 /(1 + \text{decay} * \text{epoch})$$

When the decay argument is 0, this has no effect on the learning rate and when u specific the argument it will change the learning rate by a fixed amount.

Drop-Based Learning Rate Schedule: In this approach we will be decreasing the learning rate using punctuated large drops at specific epochs. We will be implementing this by dropping the learning rate by half every fixed number of epochs. For example, if we take the initial learning rate of 0.1 and drop it by 0.5 every 10 epochs. The first 10 epochs of training would use a value of 0.1, in the next 10 epochs a learning rate of 0.05 would be used, and so on. I used LearningRateScheduler callback while fitting the model as it allows us to define to call that takes the epoch number as an argument and returns the learning rate to use in stochastic gradient descent. A new step_decay() function is defined that implements the equation:

```
LearningRate = InitialLearningRate*DropRate^floor(Epoch/EpochDrop)
```

Where InitialLearningRate is the initial learning rate such as 0.1, the DropRate is the amount that the learning rate is modified each time it is changed such as 0.5, Epoch is the current epoch number and EpochDrop is how often to change the learning rate such as 10.

While using this learning rate the learning rate specified by stochastic gradient descent is ignored and set to 0.

D. *Validation of Best Model:*

I also checked the homogeneity of the dataset using cross validation for 5-kfold and the MSE obtained was ([-0.01241179, -0.0132576, -0.01369088, -0.01290721, -0.01361953]). As we can see the range is between 0.0124 and 0.0136, the change is very insignificant hence we can fit our model in a homogeneous manner on the dataset.

I also plotted learning curve for instance size up to 200 to see if there is any pattern. The change of training and validation MSE scores does not make difference by sample size. It seems the sample size does not influence the performance of the model, so we have adequate sample size to achieve the same result. The curve keeps increasing until it converges to 0.01 as the sample size increases. We can conclude that our model is not overfitting.[8]

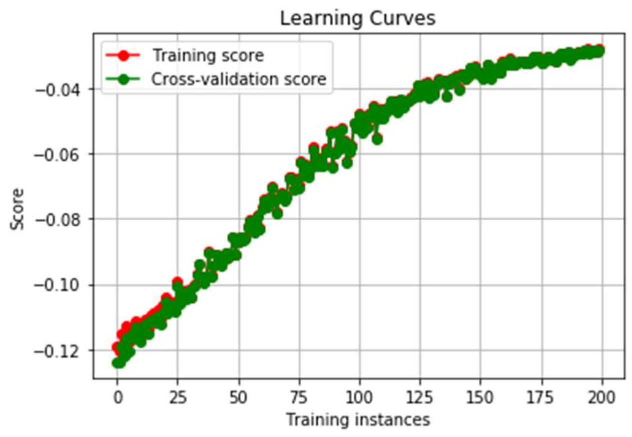

Fig. 1. Training Score vs Cross-validation score for 200 instances

E. *Results Table:*

Feature	**Spearman's correlation coefficient for each model**		
	Model Used	***Short-term***	***Long-term***
Captions	RandomForestRegressor(n=10)	0.383	0.152
	RandomForestRegressor(n=100)	0.413	0.149
	SupportVectorRegressor	0.437	0.189
	NeuralNetwork	**0.443**	0.204
Captions with weights	RandomForestRegressor(n=10)	0.398	0.119
	RandomForestRegressor(n=100)	0.430	0.144
	SupportVectorRegressor	0.392	0.183

Feature	Spearman's correlation coefficient for each model		
	Model Used	***Short-term***	***Long-term***
	NeuralNetwork	0.401	0.198
Captions & HMP	RandomForestRegressor(n=10)	0.246	0.099
	RandomForestRegressor(n=100)	0.327	0.119
	SupportVectorRegressor	0.416	0.182
	NeuralNetwork	0.407	**0.230**
Captions & C3D	RandomForestRegressor(n=10)	0.256	0.088
	RandomForestRegressor(n=100)	0.333	0.101
	SupportVectorRegressor	0.409	0.155
	NeuralNetwork	0.408	0.194

a. split random seed = 42

b.

For final submission I used captions with our baseline neural network to predict short-term memorability score and a combination of HMP features to predict the long-term memorability score. For the dummy test dataset, I got short-term memorability score 0.443 and long-term score is 0.230.

CONCLUSION *(HEADING 5)*

After working with different set of features and using both non-parametric and neural network approaches of machine learning I concluded that captions provide the best results and even combining them with video features does not give any significant improvement. Although, captions with additional weights to certain words do provide significant results. This area need more exploration and we could also check the results after assigning negative weights to certain words. The right combination of words might provide a breakthrough in this area.

REFERENCES

[1] "MediaEval 2019." http://www.multimediaeval.org/mediaeval2019/ (accessed Apr. 20, 2020).

[2] "MediaEval 2018: Predicting Media Memorability Task," *GroundAI.* https://www.groundai.com/project/mediaeval-2018-predicting-media-memorability-task/1 (accessed Apr. 20, 2020).

[3] R. Chaudhry, M. Kilaru, and S. Shekhar, "Show and Recall @ MediaEval 2018 ViMemNet: Predicting Video Memorability," p. 3.

[4] T. Joshi, S. Sivaprasad, S. Bhat, and N. Pedanekar, "Multimodal Approach to Predicting Media Memorability," p. 3.

[5] R. Cohendet, C.-H. Demarty, N. Duong, and M. Engilberge, "VideoMem: Constructing, Analyzing, Predicting Short-Term and Long-Term Video Memorability," in *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, Seoul, Korea (South), Oct. 2019, pp. 2531–2540, doi: 10.1109/ICCV.2019.00262.

[6] R. Gupta and K. Motwani, "Linear Models for Video Memorability Prediction Using Visual and Semantic Features," p. 3.

[7] J. Brownlee, "Using Learning Rate Schedules for Deep Learning Models in Python with Keras," *Machine Learning Mastery*, Jun. 21, 2016. https://machinelearningmastery.com/using-learning-rate-schedules-deep-learning-models-python-keras/ (accessed Apr. 20, 2020).

[8] J. Brownlee, "How to use Learning Curves to Diagnose Machine Learning Model Performance," *Machine Learning Mastery*, Feb. 26, 2019. https://machinelearningmastery.com/learning-curves-for-diagnosing-machine-learning-model-performance/ (accessed Apr. 20, 2020).