AD DETECTOR

Srinath Kanna Dhandapani
Kowsic Jayachandiran

Abstract

News channels on television follow a format that is different from regular television channels in that the news channels have a few sections in a video frame that keep updating news even when the advertisements are running in the background. This makes identifying if a video frame is a news video or an advertisement quite tricky. The project aims at detecting if a frame is from an advertisement or from news based on audio and video features extracted from a few English and Indian news channels. We have experimented with various classification models and the results from each are analyzed and compared with results from a previous approach that was based off of Support Vector Machines. Two of the approaches have been tried with and without audio features, resulting in surprising results as there was not much improvement in accuracy. We have also used unsupervised learning techniques to identify the labels of the data sets.

Introduction

Television channels have come a long way from being just a recorded media broadcasting device to telecasting up-to-the-minute events around the world. To provide event updates as and when they happen, television networks require funds, which they acquire using subscriptions or through advertisements. Any given network will always try to beat its competition in terms of news content and who relays news first. To provide such content, they require constantly flowing funds, one of the sources being advertisements. While regular television channels completely stop the media telecast and relay advertisements, news channels, however, provide only a part of the frame for advertisements, with the remaining part of the frame being reserved for news texts (headlines). After the advertisements complete their run, the portion of frame that had the advertisement running will be replaced by the news broadcast. This makes it quite hard to identify if a given frame is an advertisement or news.

Data

The dataset comprises of features extracted from 150 hours of broadcast videos from five different channels - CNNIBN, BBC, CNN, NDTV 24X7, TIMESNOW [3]. The features in the dataset contain information about audio and video of a particular frame - like bag of audio words, edge change ratio, text area distribution. Advertisements usually have some kind of music associated with it, which can be identified using low level audio features like Short Time Energy, Zero Crossing Rate, Spectral Centroid, Spectral Flux, Spectral Roll-Off Frequency, and Fundamental Frequency. The mean and variance are calculated for all the mentioned low level audio features, thus giving a 2D vector.

Advertisements are also very short in length with fast transitions in frames which can also help in distinguishing between an advertisement and a news video. Frame Difference Distribution is used to identify the changes in intensities between two different frames. Motion Distribution, Frame Difference Distribution and Edge Change Ratio can help identify the changes in two consecutive frames easily. Edge Change Ratio finds how the edges of the objects in the image vary between two consecutive frames. We then calculate mean and variance of Edge Change Ratio over a shot.

Related Work

While this project heavily focuses on finding how different classification models perform with the dataset, Vyas et al. [1] had worked with extracting audio and video features from 54 hours of news footages, and using SVM to classify the dataset into advertisements and non-commercial videos. The authors had obtained an F-measure of 97%. Nikola Banić proposed a method for logo detection in videos [2], thereby identifying if a video is an advertisement or news footage since commercials do not always have the channel's logo on them; this model gave an accuracy of 100% in recognizing commercials and 80% in determining the beginning and ending of a commercial.

Method

The Data Set comprised of nearly 130000 instances from five channels which were randomly split into training and testing data, allocating 70% of the available instances to training set and the remaining for testing set.

Based on the number of classes in the classification problem that this dataset deals with, decision trees are considered to be the best (as it suits well for two-class problems). In the previous approach [1], the results from Support Vector Machines (SVM) produces a F1 measure of 97%. In order to test the performance of decision trees over the same data set and determine its performance, a set of three different tree models were chosen.

We used random forest as one of our training models as it uses a collection of decision trees and predicts output based on the ranking algorithm's result from all the randomly generated decision trees from random splits in dataset. This approach is also called as bagging and it greatly improves the result of decision trees. Having the approach of random forest, it was also necessary to use decision trees as a standalone classifier and check its results. Decision trees sometimes produce similar results as random forest, as it does not have the randomness factor involved in the classification and gives the same result after each run. Adaboost trees are in turn similar to bagging logic and it tries to correct the errors of the tree by creating a separate logic to learn from the mistakes. So the approach involved the comparison of results from three different classifiers for this classification problem.

Clustering the dataset and comparing the results from unsupervised learning and the already labelled results were necessary to see how well each of the features played its role in determining the similarity of all the instances belonging to a particular class. So, K-Means was chosen for this purpose as it not only allows the option to set number of clusters manually but also gives equal importance to all its features.

Experiments

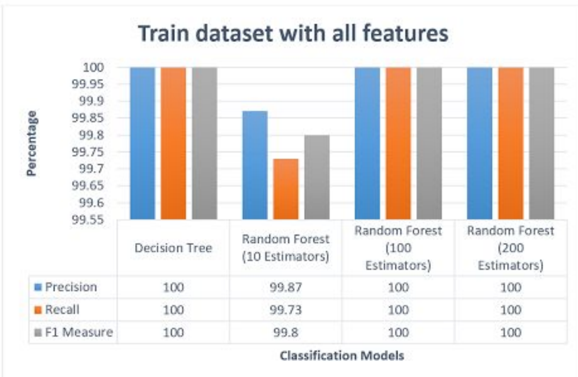

Fig. 1: Train dataset with all features

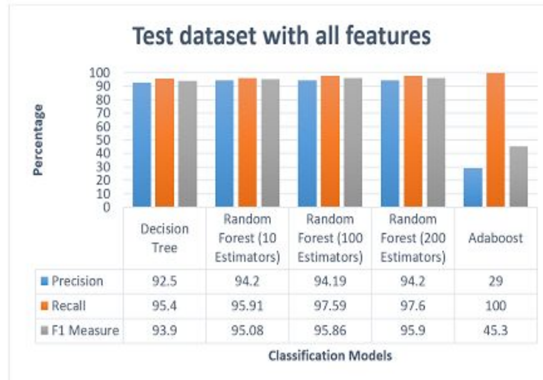

Fig. 2: Test dataset with all features

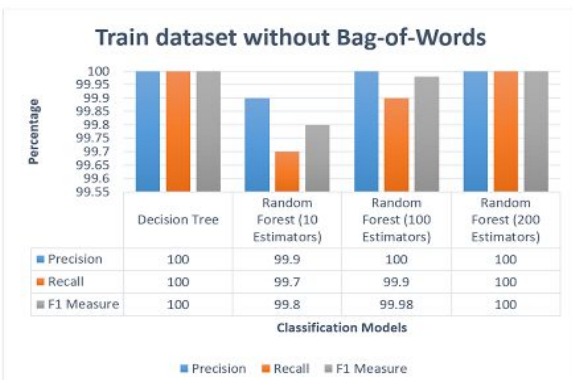

Fig. 3: Train dataset without BoW

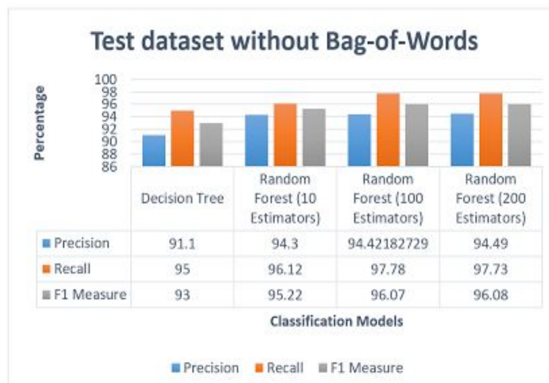

Fig. 4: Test dataset without BoW

Two types of experiments were executed, with one considering all the features in the data set and the other excluding the bag of words features. The first method dealt with 4124 features while the second method modelled the data with 124 features. The results were quite surprising as there was not much difference in the results obtained in the two cases. It significantly reduced the training time and could almost match the results. This lead us to believe that the Bag of Words features were overfitting the model without imparting any additional knowledge or patterns to classify the data.

As expected the models give better results when tested with training data. When tested, random forest gave the best results among the tested models compared to decision tree and adaboost. The models with better results were given more preference and tested with additional parameters like increasing the number of estimators (*n_estimators*). We observed that the performance kept increasing with the increase in *n_estimators* proportionally and could see a sudden decrease after the threshold of 100 and the best F1 measure was determined to be 95.9 considering all the features. The best F1 Measure of 96.08 was observed with random forest for the model with 200 *n_estimators*, by not including the Bag-of-Words features.

The degraded results from the adaboost model were due to the fact that when two models try to perform their best, they started to classify all the data as non-commercials owing to the fact that 70% of the data comprises of this class resulting in 100% recall and 29% precision. It was also analyzed by changing the adaboost classifier model that but the results were almost similarly degraded, mainly supporting the reason that adaboost does not work well for two class problems. The datasets seem to work well with random forest without much overfitting and producing almost similar results to the previous work [1].

Fig. 5: K-Means Clustering

Another objective of the project was to implement unsupervised learning to determine if it would be able to cluster the dataset into two clusters given all the features, except Bag-of-Words features, in the model, and to then verify if it was able to cluster correctly. K-Means clustering algorithm was used for this purpose. The results of K-Means algorithm were then compared with the actual data set labels and was found to have an F1 measure of 77.97. Although there are many other ways to cluster the dataset, the observed high results prove the shared similarity among the data points of the two classes.

Conclusion

The current features just works on Bag-of-Words model to deal with the audio features and it becomes very difficult and time consuming when the feature length keeps on increasing. This can be handled in a better way if there were a substitute of audio features representation with word2vec [5] and extract more advanced video visual features using computer vision techniques and deep neural networks.

References

[1] Vyas Apoorv, Raghvendra Kannao, Vineet Bhargava, and Prithwijit Guha. 2014. “Commercial Block Detection in Broadcast News Videos.” In *Proceedings of the 2014 Indian Conference on Computer Vision Graphics and Image Processing - ICVGIP ’14*. doi:10.1145/2683483.2683546.

[2] N. Banic. Detection of commercials in video content based on logo presence without its prior knowledge. In *MIPRO, 2012 Proceedings of the 35th International Convention*, pages 1713–1718, 2012.

[3] DataSet.

[4] Scikit-Learn Library.

[5] Word2Vec.