AUTOMATIC DETECTION OF VIDEOS' SCENES WITH AGGRESSION UTILIZING MOVIES' TRANSCRIPTS BY USING TEXT MINING TECHNIQUES

Badriya Murdhi Alenzi

Information Systems Department
College of Computer and Information Sciences
Al Imam Mohammad ibn Saud Islamic University(IMSIU),Riyadh, KSA

Muhammad Badruddin Khan

Information Systems Department
College of Computer and Information Sciences
Al Imam Mohammad ibn Saud Islamic University(IMSIU), Riyadh, KSA

Abstract

The world is witnessing revolutionary evolution of internet and with the advent of social media; users are empowered to easily post contents on the web at any time and from any place in the form of opinions, comments, and feelings. Manual approaches of detecting and analyzing such huge amount of posts are not feasible and there is a need for automated methods and techniques to discover the knowledge and patterns of the text content without human involvement. Text mining refers to the process of extracting interesting and significant patterns or knowledge from text documents. YouTube is known for its free provision of video sharing service. The content of YouTube videos may sometimes comprise of images or sequence(s) of images with unwanted material, such as aggression, which is the reason of emergence of many social problems, particularly among children such as demonstration of aggressive behavior and bullying at home, school and public places. The research work reports performance of machine learning classifiers that were applied on video transcripts of YouTube videos to detect aggression. The dataset constructed for the purpose of research work, consists of English video scenes transcripts that were collected from the web and were annotated manually as violent and non-violent. Various experiments were performed on the dataset using different machine learning (ML) classifiers with different text preprocessing settings in RapidMiner and Python environments and thus predictive classifier models were constructed and tested. In RapidMiner environment, the SVM classifier model outperformed the other classifiers achieving highest accuracy of 79% after preprocessing step of removal of stop words. In Python programming environment, NB classifier outperformed the other classifiers in majority of experiments with different preprocessing settings, achieving highest accuracy of 82.5%, when stemming was performed in preprocessing stage along with other preprocessing steps. The automatic process of aggression detection in video scenes can be used by concerned authorities to enforce their cultural priorities.

Article Highlights

- The text of a movie transcript is a useful source to detect aggression in movie scene. The article shows that the text mining techniques can be helpful in utilizing the movie transcript to detect violence at scene level.
- By presenting the results of different experiments, the article demonstrates that application of different text processing techniques on movie transcript can improve different classification algorithms performance to detect aggression in movie scenes.
- The article also includes comparison of results of experiments performed in two environments:

  Machine learning software (RapidMiner) where programming is almost not needed
  - Python programming environment.
  - Thus, the researchers who want to work in similar domain can decide which environment is suitable for their research purposes.

Index Terms

Video transcript, Aggression detection, Machine learning, Vector Space Model, Term Frequency- Inverse Document Frequency, Natural Language Toolkit, Decision Tree, Naïve Bayes, K-Nearest Neighbor, Support Vector Machine, Weka - RIpple-DOwn Rule learner.

I. INTRODUCTION

In recent years, with the rapid development of the information technology resources, huge amounts of text data (unstructured data) are produced at unprecedented pace using social networks and other web sources. Naturally, this progress

Manuscript received September 5, 2020
Manuscript revised September 20, 2020

DOI: 10.22937/IJCSNS.2020.20.09.24

has resulted in the increased focus of researches from academia as well as industry on the generated data. Text mining methods and algorithms can be used to detect interesting patterns automatically from the text data in a scalable and effective manner. One of the primary concerns of the web users is the availability of inappropriate and harmful content on the web [1].

YouTube is a public video sharing site, where the users can provide the video content by uploading their videos, or they can be consumers and can view, rate, share, and comment on different videos. Even in the consumer role, the activity of users is a source of creation of another form of content that is textual in nature when it is in form of comments, numeric when it is in form of rating. YouTube offers various types of videos that have different subjects including documentary films, educational videos, TV shows, movie trailers, and video clips, etc. [2]. Videos on YouTube have different types of content, which may include undesirable things such as aggression which is considered as a major reason of emergence of many social problems among children, like the issue of violence at home, in school and in public places.

This research work aims to address the problem of content that contains aggression and is freely available in online videos. The target is to detect scenes with such aggressive content in videos by using machine learning techniques and thus to enable authorities to take appropriate steps in the light of their cultural priorities. The work can even be used to empower video sharing websites to put certain restrictions on videos with aggression. This will result in safer internet and will give parents more confidence and comfort that they are being assisted by other agencies to raise their child in healthy and peaceful environment.

The definition of violence by the World Health Organization is "the intentional use of physical force or power, threatened or actual, against oneself, another person, or against a group or community, which either results in or has a high likelihood of resulting in injury, death, psychological harm, maldevelopment, or deprivation" [19]. There are few studies that focused their attention on aggression detection, but to best of our knowledge, there exist only one study that used user comments to classify movie as violent or non-violent [3]. Furthermore, there are few limited studies focusing on analysis of video transcripts for different purposes but there are no studies focusing on analyzing video transcripts for detecting aggression (video transcription is the process of translating a video's audio into text) [4].

In the experiments performed to achieve the target of this research work, machine learning classifiers in the environments of RapidMiner and Python programming language were applied on video scene transcripts to construct predictive models to classify video scenes into violent or non-violent classes automatically. The effects of text preprocessing techniques on video transcripts were examined and the classifications' performances were calculated. This paper provides details of experiments performed to achieve the target of the research.

The paper structure is as follows: Section (2) presents some of the previous studies that discussed topics related to the presented work. Section (3) exhibits the steps of the methodology used to make this paper study. Section (4) displays the results of the study. Section (5) displays the finding and some recommendations for future work.

II. RELATED WORK

III.YouTube is the most common video sharing site that have a massive number of videos are uploaded every day. There are many text mining studies applied on YouTube content. M. Wöllmer et al. [5] focused on analyzing the reviews of YouTube videos to automatically determine a speaker's feelings. L. P. Morency, R. Mihalcea, and P. Doshi [6] used multimodal sentiment analysis to classify the opinions in YouTube videos. S. Poria et al [7] presented a new methodology, which used textual, audio, and visual modalities to capturing patterns from Web videos. M. Thelwall, P. Sud, and F. Vis [8] analyzed the comments of YouTube's videos to gather the reaction of the public to some issues or special videos.

One of the essential concerns for web users is the presence of unsuitable and harmful content on the web like, aggression. Aggression is the behavior that leads to offensive against the well-being (physical, emotional, psychological) of an individual or group [26]. There are many studies that discussed the different forms of aggression. Y. Elovici et al [9] presented a system for terrorist detection that was used to monitor the traffic of specific group of users, analyzed the information that the suspected people had been accessing, and if the information was not within the group interests, the system gave an alarm. P. Calado et al [10] classified the web documents of anti-terrorism applications by combining text-based similarity metrics with link-based similarity measures. W. Warner and J. Hirschberg [11] presented an approach for detecting hate words and developed a mechanism for discovering methods used to avoid the filters of dirty words. S. Liu and T. Forss [12] used methods of topic extraction and modeling to develop a classification model used to discover intolerance, aggression, and hateful web content. D. Won, Z. C. Steinert-Threlkeld, and J. Joo [13] developed a visual model used to monitor the protesters by studying the visual attributes of images, and evaluate the level of aggression in those images.

D. A. Al Wedaah [3] used text mining techniques to analyze the comments of cartoon's movies to detect aggression. The study collected comments for 1,177 YouTube videos of cartoon and used them to build the classifiers using natural language toolkit in Python and RapidMiner. The classification algorithms such as decision trees (DTs), Support Vector Machine (SVM) and Naïve Bayes were used with preprocessing techniques in order to increase the classifiers performance. The beast accuracy 91.71% was given by NLTK and Naïve Bayes classifier with an error rate of 8.29%.

Video transcription is the process of converting a video's audio into text by using human transcriptionists, speech recognition technology, or a combination of the two [16]. There are limited studies focusing on using video transcripts in text mining. N. Sureja [14] used the subtitles of movie and movie type like action, drama, and comedy to build a model using lexical based approach. A. Blackstock and M. Spitz [15]

classified scripts of movies into different genres according to the features of natural-language processing (NLP), which were extracted from the scripts. Hence, there are few studies that analyzed video transcripts for different objectives but there are no studies, to the best of our knowledge, that were focused on violence detection in videos' scenes using videos' scenes transcripts.

This paper present research work that focuses on analyzing Anime video transcripts (new input type to be used for research) to detect aggression in YouTube movies by using the machine learning classifiers. For the purpose of the study, a corpus contains 100 Anime video scenes was constructed and these transcripts were manually labeled as violent and non-violent by the researcher and two other persons after watching their corresponding scenes.

IV. METHODOLOGY

The methodology of the study consists of three phases; the first phase describes the process of data collection and the sources that were used to collect the data. The second phase speaks about the mechanisms and techniques that used for data pre-processing. The third phase analyzes hoe the data was classified and discusses the evaluation process. In the following sub sections each phase will be discussed in details.

A. Data Collection

This phase describes how the Anime cartoons were selected from the Web (Anime, a style of Japanese animation made using drawn characters and images rather than real, aimed at adults and children). Based on the requirements of the study, three Anime cartoon series were selected. After that, YouTube was used to examine the Anime episodes, each with 20 minutes' duration, and then each episode was divided into separate scenes .The World Wide Web was used to collect the corresponding Anime transcripts of the chosen scenes and based on the video content each scene was labeled manually as violent or non-violent. Later the video transcripts were saved in Excel file with their labels. Total of 68 video transcripts of three different Anime cartoon series were used to gather 100 scene transcripts.

1) DATA CLEANING

The scenes were cleaned by deleting the data that don't have value in this analysis like character names .

Example:

Raw dialogue (before cleaning)

SHIBUIMARU: Hey, baby! Where are you going? Come and have a little fun with us.
PUNK: That's our Taku for ya. This guy can spot a hottie a mile away.
SHIBUIMARU: What's up, little lady? The name's Takuo Shibuimaru. What do you say? Come hang out with us, pretty lady.
WOMAN: Please. I don't want any trouble.
PUNK: You hear that? She doesn't want trouble.

Cleaned dialogue:

Hey, baby! Where are you going? Come and have a little fun with us.
That's our Taku for ya. This guy can spot a hottie a mile away.
What's up, little lady? The name's Takuo Shibuimaru. What do you say? Come hang out with us, pretty lady.
Please. I don't want any trouble.
You hear that? She doesn't want trouble.

2) SCENES ANNOTATION

The researcher and two other persons labeled the scenes manually into two classes: violent or non-violent scene based on the majority opinion after viewing the scenes not based on the text of the scene transcript. Then the scenes were updated in excel sheet as a raw text. The corpus includes 50 scenes labeled as violent and 50 scenes labeled as non-violent.

There are some problems faced during data collection process. The first problem was that the process of searching the web for suitable Anime transcripts was a time-consuming process. It needed to compare the Anime video with its corresponding transcript text to ensure that they were similar. Then, based on the watching of the movie, the transcript was divided into several scenes. The second problem was the cleaning process of the corpus. The third problem was the complex process of the scene labeling as violent or non-violent. For example, the scene describes news presenter sitting in TV office that tells news about some violence. Hence the transcript of a scene will contain violent words that will mislead machine learning classifier to assume that the scenes contain violent images even though in the real movie scene content, there will be no aggression. The three annotators followed some rules to label a scene as violent or non-violent. The rules were as follows:

1- The scenes that included sequence of images with shedding of blood and fighting with some weapons were labeled as violent.
2- The scenes that included bully actions (someone who frightens or hurts someone else) were labeled as violent.
3- The scenes that included characters with frightening shapes like a monster doing some harmful acts were labeled as violent.

B. Data Preprocessing

The pre-processing phase reduces the noise in the scenes text to improve the classification's performance. There are many preprocessing techniques available to different extents in RapidMiner and Python environments like, tokenization,

punctuation and stop words removal, and stemming that were used in our experiments.

Data Representation

After pre-processing the text, every preprocessed scene's transcript was transformed as a vector in a vector space, and for this purpose, VSM (a vector space model) was used [21]. Thus VSM represented the scenes as a bag of words so that different data mining algorithms can be applied. As shown in Figure 1, each scene was represented by keyword vectors in RapidMiner environment. Similar approach was adopted for Python Programming language environment. The reason why Python environment was used in addition to RapidMiner environment is the flexibility that Python programming environment provides, in terms of preprocessing.

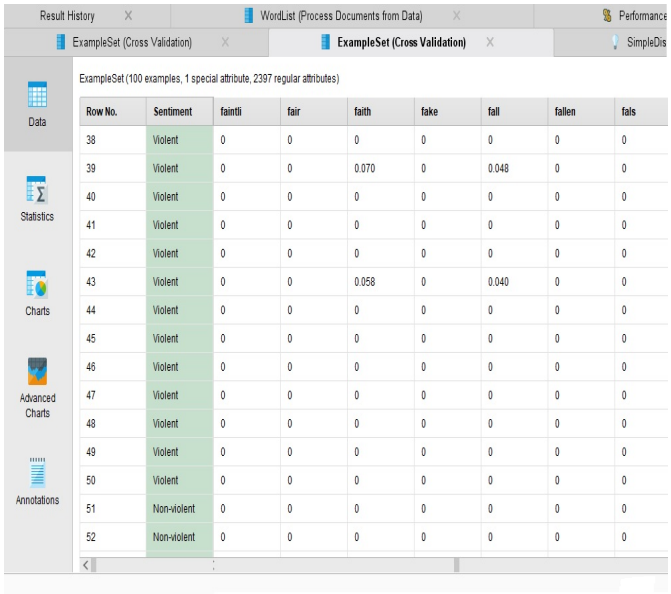

Figure 1: The representation of scene as keyword vectors in RapidMiner environment

In the generated matrix, the rows' vectors (movie scenes words) are mapped to their corresponding classes (violent or non-violent), the dimension refers to the term (word) from the scenes, and the cells represent the weight (TF-IDF) of each word. If the weight equals zero that means the word does not exist in the scene, if the weight is greater than zero that means the word exists in the scene. The Term Frequency- Inverse Document Frequency (TF-IDF) was chosen that is a numerical statistic method that produces a composite weight for each term in each document to reflect the importance of a word in a collection or corpus to a document.

The formula of the TF-IDF weighting scheme assigns to the term t, in document d, given by:

$$\text{TF-IDF}(t,d) = \text{TF}(t,d) * \text{IDF}(t)$$

Equation 1: TF-IDF Equation [18].

The term frequency is the number of occurrence of t in d, given by:

$$\text{TF}(d,t) = \text{count}(t,d)$$

Equation 2: TF Equation [18].

Where the term is t and d is the document.

The inverse document frequency of t is given by:

$$\text{IDF}(t) = \log N / \text{DF}(t)$$

Equation 3: IDF Equation [18].

Where N is the number of documents in a corpus and DF is number of the document in the corpus that contains a term t [18].

C. Classification

This work aims to use machine learning classifiers to classify movie scenes' transcripts into violent or non-violent class. This section explains the classification processes of the study work.

Machine Learning Classifiers

We used supervised learning approach also known as the corpus-based approach. It is where ML classifiers such as Decision Tree (D-Tree), Support Vector Machine (SVM), K-Nearest Neighbor (KNN), Naïve Bayes (NB) etc., are applied to a manually annotated dataset. The dataset was split into a training set and a testing set. The classifier learns from the training data set and builds a model that can be used to classify the testing dataset and the accuracy is calculated for constructed model [20].

1) RAPIDMINER MACHINE LEARNING CLASSIFIERS

RapidMiner, one of the leading open-source machine learning software, was the first system used to apply the Machine Learning algorithms in this research work. RapidMiner provides a separate parameter for the classification algorithms like Decision Tree, Naïve Bayes, Support vector machine, K-NN, W-Ridor, and W-JRip. The input of these classifiers is a set of preprocessed scenes which forms the training data. The 10-fold cross-validation method was used, which splits the dataset into nine random parts for training, and one part for testing [17].

Decision tree

A decision tree classifies the scene by starting at the root node of the tree, which is the decision or test point, and moving down the branches which represent the outcome of a decision (the TF-IDF of this word) and further down to the leaf node, which represents the classification class. Figure 2 represents an example of DT classifier.

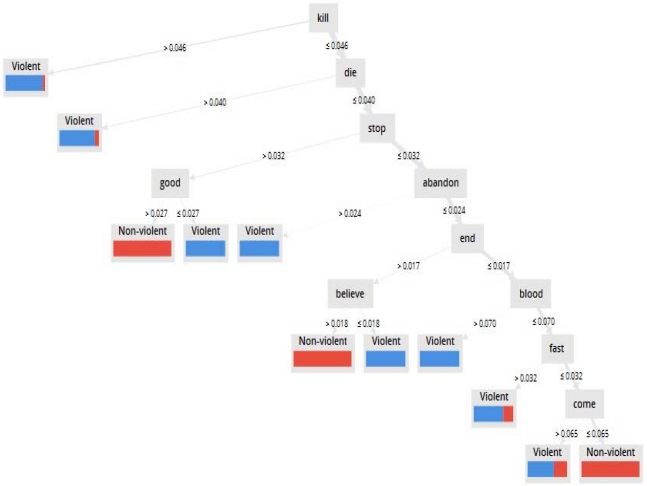

Figure 2: Representation of an example of decision tree for video scenes transcripts classification.

Naive Bayes (NB) Classifier

It is a probabilistic classifier based on Bayes' Law and naïve conditional independence assumptions. It assumes that the presence (or absence) of a particular feature of a class is independent to the presence (or absence) of any other feature. It considers each word in a scene as an independent word which means that there are no dependences between the words in the training data [18]. We used Weka's Naïve Bayes method in RapidMiner environment by installing Weka extension. Figure 3 shows the conditional class probabilities of the few words in the dataset.

W-NaiveBayesMultinomial

```
The independent probability of a class
--------------------------------------
Violent 0.5
Non-violent     0.5

The probability of a word given the class
-----------------------------------------
         Violent Non-violent
jealou  3.773558294523169 5E-4    5.200550991377859E-4
jean    4.941218377505395E-4      7.398163140925 03E-4
jerk    4.426704667821982E-4      4.448092275542327E-4
join    4.020995987641838 5E-4    5.631332485154 96E-4
joke    3.989893767633831 25E-4   4.373445474320372E-4
jose    4.140162267048806E-4      3.762750495745783E-4
journei 4.136357399266205E-4      3.762750495745783E-4
judg    3.900768735649531E-4      4.627939366949577E-4
judgement       3.773558294523169 5E-4    4.861414324894307 6E-4
judgment        4.485671705353224E-4      4.445790613236518 5E-4
jump    5.041608240886827E-4      4.259325943657076 5E-4
jumpi   3.773558294523169 5E-4    4.328413190370239E-4
junki   4.033415859205208E-4      4.316745220214604 6E-4
justadd 3.773558294523169 5E-4    3.953229743784926 4E-4
justic  4.304728936975727E-4      3.762750495745783E-4
justifi 3.773558294523169 5E-4    4.439127562794129E-4
kakuzawa        4.145460081475748E-4      5.133434055338869E-4
kamakura        4.234068937320025 5E-4    3.762750495745783E-4
kana    3.773558294523169 5E-4    5.683067880272 21E-4
keehl   4.140162267048806E-4      3.762750495745783E-4
keen    4.154665747412397 6E-4    3.762750495745783E-4
keep    7.876199329598 36E-4      5.526604907726606E-4
keith   3.773558294523169 5E-4    4.395662488160904 7E-4
```

Figure 3: Representation of the probabilities for few words in Naïve Bayes classifier.

Support Victor Machine Classifier

SVM is a supervised machine learning algorithm which can be used for both regression and classification challenges. It is mostly used in classification problems. Given a set of training data, where each scene vector belongs to one of two categories (violent or non-violent), an SVM builds a model that assigns the new dataset to one category which makes it a non-probabilistic binary linear classifier. SVM model represents the scene vectors as mapped points in space so that they are divided by a clear gap that is as wide as possible based on its separate category. Then new scene vectors are mapped into that same space and based on which side of the gap they fall, their category is predicted [22].

K-Nearest Neighbor (K-NN) Classifier

K- Nearest Neighbors algorithm (K-NN) is a non-parametric algorithm used for classification, the input of this algorithm is k which is a positive small integer and the output is a class membership (k is the closest training examples in the feature space). K-NN algorithm is one of the simplest machine learning algorithms; the scene vector is classified based on the majority of its neighbors. If k = 3, then the scene vector is simply assigned to the nearest three neighbors' class [23].

RIpple-DOwn Rule Learner Classifier

It is a direct classification method. It generates the default rule and finds exceptions of that default rule with the smallest error rate by using the incremental reduced error pruning. Then, it finds the best exceptions for each exception, and iterating until pure. A tree-like expansion of exceptions produces the most excellent exceptions which are created by each of the exceptions. The exceptions are the rules which predict classes other than the default. Exceptions are created by using the incremental reduced pruning algorithm (IREP) [24]. We used Weka's Ridor method in RapidMiner environment by installing Weka extension. Figure 4 shows the representation of the W-Ridor classifier for the scenes.

W-Ridor

```
RIpple DOwn Rule Learner(Ridor) rules
-------------------------------------

Sentiment = Non-violent  (100.0/50.0)
          Except (hell > 0.024912) => Sentiment = Violent  (19.0/3.0) [8.0/4.0]
          Except (fight > 0.032554) => Sentiment = Violent  (11.0/1.0) [4.0/1.0]

Total number of rules (incl. the default rule): 3
```

Figure 4: Representation of the W-Ridor classifier.

W-JRip Classifier

JRip is an inference and rule-based learner, called Repeated Incremental Pruning to Produce Error Reduction (RIPPER) that tries to find propositional rules that can be used to classify elements. It was suggested by William W. Cohen as a developed version of IREP. It uses word-based dataset to produce a set of rules that contains both expected rules and sometimes unexpected rules that identify the classes while minimizing the amount of error. It is based on association rules with reduced error pruning (REP), which is an effective technique of decision tree algorithms. In REP, the training data is divided into a growing set and a pruning set. First, an initial set of rules is formed based on the growing set by using heuristic method. Then, the initial rule set is simplified repeatedly by applying one of the pruning operators which delete any single condition or

rule. At each stage of simplification, the selected pruning operator is the one that gives the greatest error reduction on the pruning set. Simplification ends when using any pruning operator would increase error on the pruning set [25]. We used Weka's Ripper method in RapidMiner environment by installing Weka extension. Figure 5 shows the representation of the W-JRip classifier for the scenes.

W-JRip

```
JRIP rules:
===========

(kill >= 0.021413) => Sentiment=Violent (31.0/8.0)
(alive >= 0.016311) => Sentiment=Violent (8.0/1.0)
(end >= 0.033007) => Sentiment=Violent (11.0/3.0)
(come >= 0.084773) => Sentiment=Violent (2.0/0.0)
 => Sentiment=Non-violent (48.0/10.0)

Number of Rules : 5
```

Figure 5: Representation of the W-JRip classifier.

2) PYTHON MACHINE LEARNING CLASSIFIERS

For this research work, Scikit-Learn library was used, which is a free software machine learning library for Python programming language. It applied various classification algorithms like Multinomial NB, SVM, Decision Tree, and K-NN. The English were read from Excel file which contains the scenes before and after preprocessing.

Training and Testing Phase:

The Cross-Validation method was used which split the dataset into training set and testing set. 10-fold cross-validation was used, which splits the dataset into nine random parts for training, and one part for testing. In the training phase, features were extracted by extracting every word in the scenes of the training set followed by creation of a vector and returning of TF-IDF (The Term Frequency- Inverse Document Frequency), which reflected the word's importance in a corpus. The ML classifiers that were implemented are NB, DT, SVM, and K-NN to construct the predictive model. As previously mentioned, the NLTK tool was used to run the classifier and then to test it using the cross-validation method. The results will be discussed in the section 4 in detail. However, an example of predictive model built using decision tree classifier in Python environment is presented in Figure 6 before commencement of next section in order to show flexibility and functionality of Python programming language.

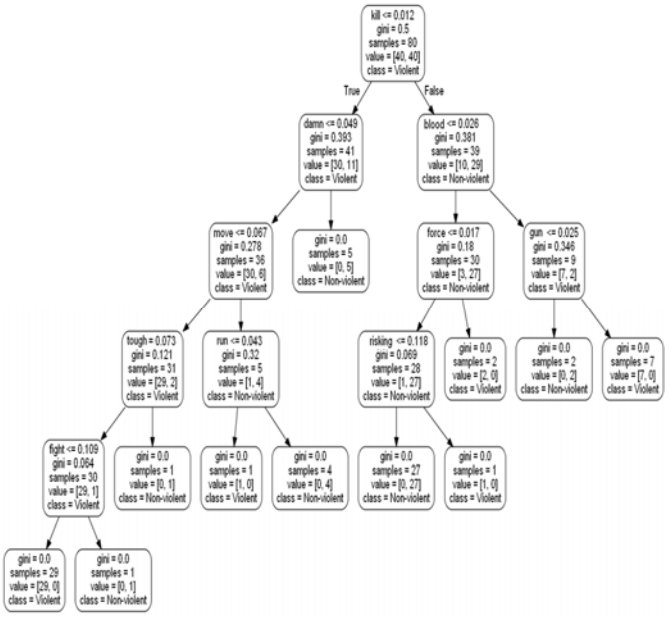

Figure 6: Representation an example of decision tree classifier.

V. RESULTS AND DISCUSSION

This section will present the results of different experiments performed in RapidMiner and Python environments or systems. We will first reiterate our objective and describe the followed method. Then we will provide results of the experiments with different preprocessing settings followed by brief discussion based on achieved performances. Hence objective, method, results and discussion will be stated for experiments in both RapidMiner and Python environment.

A. Experiments using RapidMiner Environment

1) OBJECTIVE

The objective of the experiments is to build classification models using classification methods, such as DT, NB, SVM, K-NN, W-Ridor, and W-JRip, that can automatically label video scene as violent or non-violent and to check their performance by using RapidMiner. The experiments were performed with different preprocessing settings.

2) METHOD

The dataset consists of 100 scenes transcripts, 50 violent scenes and 50 non-violent scenes, which were annotated manually. The preprocessing operators applied on the video transcripts were started by tokenization of the scenes followed by transformation of them to lower case, removal of stop words, application of stemming, and finally filtering of words by length, that is, removal of words that were less than 2 letters and more than twenty letters in our experiments. The Tables and Graphs in the results section showed the results of ML classifiers that were obtained before and after the preprocessing steps were applied cumulatively. In addition, the performance measures were calculated using 10-fold cross-validation method.

3) RESULTS

The following Tables and Graph indicate the results of different experiments performed on the dataset in Rapid Miner environment after different preprocessing stages.

Tokenization		Result			
		Precision	Recall	F-Measure	Accuracy
Decision Tree	violent	48.84%	42.00%	45.16%	49.00%
	non-Violent	49.12%	56.00%	52.33%	
W-Naïve Bayes	violent	75.56%	68.00%	71.58%	73.00%
	non-Violent	70.91%	78.00%	74.29%	
SVM	violent	72.73%	80.00%	76.19%	75.00%
	non-Violent	77.78%	70.00%	73.69%	
K-NN	violent	65.38%	68.00%	66.66%	66.00%
	non-Violent	66.67%	64.00%	65.31%	
W-Ridor	violent	56.86%	58.00%	57.42%	57.00%
	non-Violent	57.14%	56.00%	56.56%	
W-JRip	violent	57.41%	62.00%	59.62%	58.00%
	non-Violent	58.70%	54.00%	56.25%	

Table 1: The results of ML classifiers after step of tokenization on the dataset.

Tokenization + Transformation of cases		Result			
		Precision	Recall	F-Measure	Accuracy
Decision Tree	violent	62.75%	64.00%	63.37%	63.00%
	non-Violent	63.27%	62.00%	62.63%	
W-Naïve Bayes	violent	75.00%	72.00%	73.47%	74.00%
	non-Violent	73.08%	76.00%	74.51%	
SVM	violent	75.44%	86.00%	80.37%	79.00%
	non-Violent	83.72%	72.00%	77.42%	
K-NN	violent	66.07%	74.00%	69.81%	68.00%
	non-Violent	70.45%	62.00%	65.96%	
W-Ridor	violent	63.41%	52.00%	57.14%	61.00%
	non-Violent	59.32%	70.00%	64.22%	
W-JRip	violent	55.56%	50.00%	52.63%	55.00%
	non-Violent	54.55%	60.00%	57.15%	

Table 2: The results of ML classifiers after steps of tokenization and transformation to lower case on the dataset.

Tokenization + Transform cases + Stop words removal		Result			
		Precision	Recall	F-Measure	Accuracy
Decision Tree	violent	64.44%	58.00%	61.05%	63.00%
	non-Violent	61.82%	68.00%	64.76%	
W-Naïve Bayes	violent	67.92%	72.00%	69.90%	69.00%
	non-Violent	70.21%	66.00%	68.04%	
SVM	violent	75.00%	84.00%	79.25%	78.00%
	non-Violent	81.82%	72.00%	76.60%	
K-NN	violent	63.79%	74.00%	68.52%	66.00%
	non-Violent	69.05%	58.00%	63.04%	
W-Ridor	violent	53.66%	44.00%	48.35%	53.00%
	non-Violent	52.54%	62.00%	56.88%	
W-JRip	violent	58.70%	54.00%	56.25%	58.00%
	non-Violent	57.41%	62.00%	59.62%	

Table 3: The results of ML classifiers after steps of tokenization, transformation to lower cases, and stop words removal on the dataset.

Tokenization + Transform cases+ Stop words removal + Stemming		Result			
		Precision	Recall	F-Measure	Accuracy
Decision Tree	violent	59.18%	58.00%	58.58%	59.00%
	non-Violent	58.82%	60.00%	59.40%	
W-Naïve Bayes	violent	72.55%	74.00%	73.27%	73.00%
	non-Violent	73.47%	72.00%	72.73%	
SVM	violent	75.00%	84.00%	79.25%	78.00%
	non-Violent	81.82%	72.00%	76.60%	
K-NN	violent	66.07%	74.00%	69.81%	68.00%
	non-Violent	70.45%	62.00%	65.96%	
W-Ridor	violent	62.50%	50.00%	55.56%	60.00%
	non-Violent	58.33%	70.00%	63.63%	
W-JRip	violent	70.59%	72.00%	71.29%	71.00%
	non-Violent	71.43%	70.00%	70.71%	

Table 4: The results of ML classifiers after steps of tokenization, transformation to lower case, stop words removal, and stemming on the dataset.

Tokenization + Transform Cases+ Stop words removal + Stemming + Filter tokens (by length)		Result			
		Precision	Recall	F-Measure	Accuracy
Decision Tree	violent	58.00%	58.00%	58.00%	58.00%
	non-Violent	58.00%	58.00%	58.00%	
W-Naïve Bayes	violent	72.55%	74.00%	73.27%	73.00%
	non-Violent	73.47%	72.00%	72.73%	
SVM	violent	72.88%	86.00%	78.90%	77.00%
	non-Violent	82.93%	68.00%	74.73%	
K-NN	violent	66.67%	72.00%	69.23%	68.00%
	non-Violent	69.57%	64.00%	66.67%	
W-Ridor	violent	65.00%	52.00%	57.78%	62.00%
	non-Violent	60.00%	72.00%	65.45%	
W-JRip	violent	67.44%	58.00%	62.36%	65.00%
	non-Violent	63.16%	72.00%	67.29%	

Table 5: The results of ML classifiers after steps of tokenization, lower case transformation, stop words removal, stemming, and filter tokens (by length) on the dataset

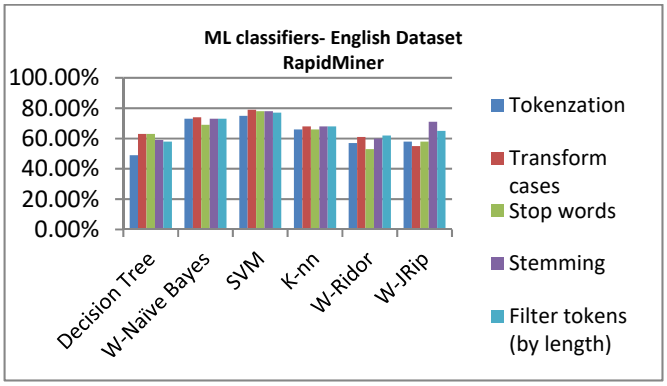

Figure 7: Comparison of the accuracies of different ML classifiers that were generated for and applied on the dataset.

4) DISCUSSION

The results of the ML classifiers performance with and without using the preprocessing operators are shown in a detailed manner in the previous section using different Tables and Figures.

It is obvious that the SVM classifier results were better than the other classifiers results with all different combinations of the preprocessing operators. The results range from 75.00% accuracy to 79.00% accuracy in different preprocessing settings. It should be note that preprocessing steps after transformation of tokens to lower case, only resulted in deterioration of the accuracy percentage. Naïve Bayes classifier competed with SVM in terms of performance in some preprocessing. From Figure 7, it can be seen that addition of step of removal of stop words in preprocessing stage has negatively impacted the performance for most of classification methods.

The rule-based classifiers such as Ridor and JRip and decision tree provide extra advantage in addition to prediction of classes. The advantage is that their models are readable or understandable. We can find which words are more related to aggression. Violent-words dictionary or lexicon can be formulated using such models. However, in the experiments, these readable classifiers were unable to provide good performances. Only in one setting, Ripper algorithm was able to reach accuracy of 71%. More insights can be gained by carefully exploring why most of the preprocessing steps resulted in deterioration of performances.

A decision tree categorizes the scene by starting at the decision node, root node, of the tree. The decision node or root node represents the word of the scene with the highest Gini Impurity or Information Gain; these are indices to measure degree of impurity quantitatively, as splitting criterion, and moving down the branches which represent the value of these words, and further down to the leaf node, which represents the classification class. In Figure 2 “Kill” is the decision node with the highest Gini Impurity [27].

B. Experiments in Python environment

1) OBJECTIVE

The objective of the experiments is to build classification models using classification methods, such as using classification methods, DT, NB, SVM, and K-NN, that can automatically label video scenes as violent or non-violent and to check their performance by using Python programming language. The experiments were performed with different preprocessing settings.

2) METHOD

The dataset used in this experiment was same as the dataset used in Experiments performed in Rapid Miner environment. The performance measures were obtained with different settings of preprocessing features: tokenization, punctuation removal, stop words removal, and Porter stemming. The Tables and Figure in the results section indicated the results of ML classifiers that were acquired when different preprocessing steps were applied cumulatively. The performance measures were calculated using 10-fold cross-validation method.

3) RESULTS

The following Tables and Figure indicate the results of different experiments performed on English dataset in Python environment after different preprocessing stages.

Tokenization		Result			
		Precision	Recall	F-Measure	Accuracy
Decision Tree	violent	55.00%	60.00%	57.00%	70.00%
	non-Violent	56.00%	50.00%	53.00%	
W-Naïve Bayes	violent	62.00%	50.00%	56.00%	81.25%
	non-Violent	58.00%	70.00%	64.00%	
SVM	violent	55.00%	60.00%	57.00%	75.00%
	non-Violent	56.00%	50.00%	53.00%	
K-NN	violent	43.00%	30.00%	35.00%	68.75%
	non-Violent	46.00%	60.00%	52.00%	

Table 6: The results of ML classifiers after tokenization on the dataset.

Tokenization + Punctuation & Stop words removal		Result			
		Precision	Recall	F-Measure	Accuracy
Decision Tree	violent	88.00%	70.00%	78.00%	68.75%
	non-Violent	75.00%	90.00%	82.00%	
W-Naïve Bayes	violent	64.00%	40.00%	67.00%	77.50%
	non-Violent	67.00%	70.00%	63.00%	
SVM	violent	56.00%	60.00%	53.00%	80.00%
	non-Violent	55.00%	60.00%	57.00%	
K-NN	violent	57.00%	40.00%	47.00%	62.50%
	non-Violent	54.00%	70.00%	61.00%	

Table 7: The results of ML classifiers after tokenization, punctuation & Stop words removal on the dataset.

Tokenization + Punctuation & Stop words removal + Stemming		Result			
		Precision	Recall	F-Measure	Accuracy
Decision Tree	violent	58.00%	70.00%	64.00%	73.75%
	non-Violent	62.00%	50.00%	56.00%	
W-Naïve Bayes	violent	64.00%	70.00%	67.00%	82.50%
	non-Violent	67.00%	60.00%	63.00%	
SVM	violent	55.00%	60.00%	57.00%	78.75%
	non-Violent	56.00%	50.00%	53.00%	
K-NN	violent	33.00%	20.00%	25.00%	68.75%
	non-Violent	43.00%	60.00%	50.00%	

Table 8: The results of ML classifiers after tokenization, punctuation & stop words removal, and stemming on the dataset.

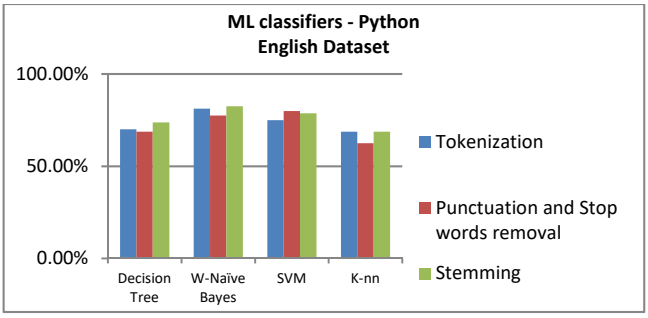

Figure 8: Comparison of accuracies of different ML classifier models that were generated in different preprocessing settings.

4) DISCUSSION

The results of the ML classifiers performance under different preprocessing settings are shown in a detailed manner in the previous section using different tables and graphs.

It is obvious that the NB classifier results outperformed other classifiers results with only one exception when SVM provided better accuracy (80%) as compared to 77.50% accuracy of Naïve Bayes. The stop word removal step had bad impact on Naïve Bayes classifier performance. The same phenomenon was observed in RapidMiner environment. SVM showed improved performance when stop words tokens were removed. This is strange because in RapidMiner environment, SVM performance was slightly degraded. The reason may be the inclusion of punctuation with stop words removal step. Again readable classifier like Decision Tree was unable to provide good performance.

C. Comparison between RapidMiner and Python Results

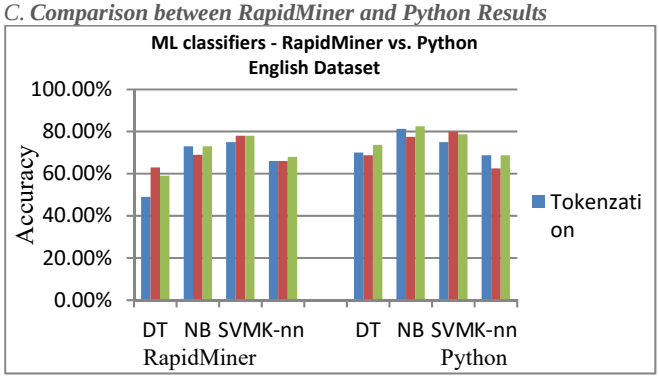

Figure 9: Comparison of accuracies of different ML classifiers results by using RapidMiner and Python that were generated for and applied on the dataset.

The outcomes after conducting the experiments by using both RapidMiner ML classifiers and Python ML classifiers based on Figure 9 are as follows:

1- The Python-based ML classifiers' performance results were better than the results using the RapidMiner platform due to better preprocessing because of programming flexibility.
2- The classification results of SVM were better than the other classifiers' results in RapidMiner environment, while the classification results of NB were better than the other classifiers' results using Python settings. The results of SVM classifier built in Python, were better than results of SVM classifier constructed using RapidMiner.
3- It was faster and easier to implement the process and obtain the results by using RapidMiner as compared to Python due to many reasons:
    a. Python is a programming language used for data mining tasks; it needs very detailed and step-by-step instructions of what to do.
    b. It requires memorizing the programming statements of the language by human to work on it.
    c. It consists of add-on packages written by others to minimize the programming effort, yet, even with the use of packages, it still requires extra efforts and some more programming.

On the contrary:

1- RapidMiner is a Graphical User Interface that tries to give the power and flexibility of programming without needing to know how to program.
2- Its workflow style is easy to use by dragging and dropping icons onto a drawing window which represent steps of the analysis. The work of the icon is controlled by dialog boxes rather than writing commands.
3- After finishing the tasks, RapidMiner documents the work steps for reusing, shows a big picture of what was done, and allows reusing the steps on new datasets without writing any programming code.

VI. CONCLUSION

There are numerous researches that have examined the harmful and inappropriate content on the web, like aggression but to best of our knowledge, video scene transcripts were not used for the purpose of violence detection. The main goal of the presented research was to detect violence in a movie by using text mining techniques so that the task can be accomplished without human intervention. For this purpose, dataset was constructed by first connecting the video scene to the video transcript followed by manual labeling of video transcripts corresponding to identified-scenes. Different data mining techniques were used on dataset with different preprocessing settings and different performance metrics were reported. Since dataset was balanced, the accuracy was one of the major performance metrics to be judged. Readable or understandable classification methods were unable to provide good performances in the experiments whereas black-box classifier like SVM showed superior performance. The best accuracy result that was achieved was 82.5% in Python environment using Naïve Bayes classifier. One way to enhance the research in future is that the special purpose violence-dictionary lexicons should be constructed and used in experiments to detect violence in movie scenes.

Acknowledgement

Authors are thankful to the annotators of the dataset that was developed for the purpose of work presented in the research article.

On behalf of all authors, the corresponding author states that there is no conflict of interest.

References

[1] Han, J., Pei, J., & Kamber, M. (2011). Data mining: concepts and techniques. Elsevier.
[2] Alexa (March 18, 2018), "youtube.com Traffic Statistics".
[3] AlWedaah D. (2015). Detecting aggression in YouTube videos using text mining techniques.
[4] GRIFFIN E. (February 3, 2015). "3 Reasons Why You Need Video Transcription.
[5] Wöllmer, M., Weninger, F., Knaup, T., Schuller, B., Sun, C., Sagae, K., & Morency, L. P. (2013). Youtube movie reviews: Sentiment analysis in an audio-visual context. IEEE Intelligent Systems, 28(3), 46-53.
[6] Morency, L. P., Mihalcea, R., & Doshi, P. (2011, November). Towards multimodal sentiment analysis: Harvesting opinions from the web. In Proceedings of the 13th international conference on multimodal interfaces (pp. 169-176). ACM.
[7] Poria, S., Cambria, E., Howard, N., Huang, G. B., & Hussain, A. (2016). Fusing audio, visual and textual clues for sentiment analysis from multimodal content. Neurocomputing, 174, 50-59.
[8] Thelwall, M., Sud, P., & Vis, F. (2012). Commenting on YouTube videos: From Guatemalan rock to el big bang. Journal of the American Society for Information Science and Technology, 63(3), 616-629.
[9] Elovici, Y., Shapira, B., Last, M., Zaafrany, O., Friedman, M., Schneider, M., & Kandel, A. (2005, May). Content-based detection of terrorists browsing the web using an advanced terror detection system (ATDS). In International Conference on Intelligence and Security Informatics (pp. 244-255). Springer, Berlin, Heidelberg.
[10] Calado, P., Cristo, M., Gonçalves, M. A., de Moura, E. S., Ribeiro-Neto, B., & Ziviani, N. (2006). Link-based similarity measures for the classification of Web documents. Journal of the American Society for Information Science and Technology, 57(2), 208-221.
[11] Warner, W., & Hirschberg, J. (2012, June). Detecting hate speech on the World Wide Web. In Proceedings of the second workshop on language in social media (pp. 19-26). Association for Computational Linguistics.
[12] Liu, S., & Forss, T. (2015, November). New classification models for detecting Hate and Violence web content. In 2015 7th International Joint Conference on Knowledge Discovery, Knowledge Engineering and Knowledge Management (IC3K) (Vol. 1, pp. 487-495). IEEE.
[13] Won, D., Steinert-Threlkeld, Z. C., & Joo, J. (2017, October). Protest activity detection and perceived violence estimation from social media images. In Proceedings of the 25th ACM international conference on Multimedia (pp. 786-794). ACM.
[14] Sureja N. (2016). A Review on Movie Script Classification using Sentimental Analysis Approach.
[15] Blackstock, A., & Spitz, M. (2008). Classifying movie scripts by genre with a MEMM using NLP-Based features.
[16] Denis, A., Cruz-Lara, S., Bellalem, N., & Bellalem, L. (2014, June). Visualization of affect in movie scripts. In Empatex, 1st International Workshop on Empathic Television Experiences at TVX 2014.
[17] Verma, T., Renu, R., & Gaur, D. (2014). Tokenization and filtering process in RapidMiner. International Journal of Applied Information Systems, 7(2), 16-18.
[18] EMC. (2013). data Science and Big Data Analytics Student Guide.
[19] Krug, E. G., Mercy, J. A., Dahlberg, L. L., & Zwi, A. B. (2002). The world report on violence and health. The lancet, 360(9339), 1083-1088.
[20] Hamouda, A., & Rohaim, M. (2011, January). Reviews classification using sentiwordnet lexicon. In World congress on computer science and information technology (Vol. 23, pp. 104-105). sn.
[21] Turney, P. D., & Pantel, P. (2010). From frequency to meaning: Vector space models of semantics. Journal of artificial intelligence research, 37, 141-188.
[22] Ray S. (2017). Understanding Support Vector Machine algorithm from examples.
[23] Altman, N. S. (1992). An introduction to kernel and nearest-neighbor nonparametric regression. The American Statistician, 46(3), 175-185.
[24] Veeralakshmi, V., & Ramyachitra, D. (2015). Ripple down rule learner (ridor) classifier for iris dataset. Issues, 1(1), 79-85.
[25] Cohen W. (1995). Fast effective rule induction. In Machine Learning Proceedings: Elsevier, pp. 115-123.
[26] Herring S. C. (2002). Gender Aggression: Recognizing and Resisting Abuse in Online Environment. Asian women, 14, 187-212.
[27] Baradwaj, B. K., & Pal, S. (2012). Mining educational data to analyze students' performance. arXiv preprint arXiv:1201.3417.

Badriya M Alenzi received the MS. degree in Information Systems from Al-Imam Mohammed Bin Saud Islamic University, Riyadh, KSA in 2017. She is currently serving as lecturer in Information System at Imam Mohammed Bin Saud Islamic University, Riyadh, Saudi Arabia.

Dr. Muhammad Badruddin Khan obtained his doctorate in 2011 from Tokyo Institute of Technology, Japan. He is a full-time assistant professor in department of Information Systems of Al-Imam Muhammad Ibn Saud Islamic University since 2012. The research interests of Dr. Khan lie mainly in the field of data and text mining. He is currently involved in number of research projects related to machine learning and Arabic language including pandemics prediction, Arabic sentiment analysis, improvement of Arabic semantic resources, Stylometry, Arabic Chatbots, trend analysis using Arabic Wikipedia, Arabic proverbs classification, cyberbullying and fake content detection, and violent/non-violent video categorization using Youtube video content and Arabic comments, and has published number of research papers in various conferences and journals. He is also co-author of a book on machine learning.