MULTILABEL MOVIE GENRE CLASSIFICATION FROM MOVIE SUBTITLE USING SUPERVISED AND UNSUPERVISED MACHINE LEARNING APPROACH

BY

MD. MEHEDI HASAN
ID: 171-15-9021

SUSANTA CHANDRA DEBNATH
ID: 171-15-9093

AND

MD. MOZAHID HASAN
ID: 171-15-9554

This Report Presented in Partial Fulfillment of the Requirements for the Degree of Bachelor of Science in Computer Science and Engineering

Supervised By

Aniruddha Rakshit
Senior Lecturer
Department of CSE
Daffodil International University

Co-Supervised By

Md. Zahid Hasan
Assistant Professor
Department of CSE
Daffodil International University

DAFFODIL INTERNATIONAL UNIVERSITY

DHAKA, BANGLADESH

JUNE 2021

APPROVAL

This Project titled “**Multilabel Movie Genre Classification from Movie Subtitle Using Supervised and Unsupervised Machine Learning Approach**”, submitted by **Md. Mehedi Hasan**, ID No: **171-15-9021**, **Susanta Chandra Debnath**, ID No: **171-15-9093** and **Md. Mozahid Hasan**, ID No: **171-15-9554** to the Department of Computer Science and Engineering, Daffodil International University, has been accepted as satisfactory for the partial fulfillment of the requirements for the degree of B.Sc. in Computer Science and Engineering and approved as to its style and contents. The presentation has been held on 2 June 2021.

BOARD OF EXAMINERS

Dr. Touhid Bhuiyan **Chairman**
Professor and Head
Department of CSE
Faculty of Science & Information Technology
Daffodil International University

Subhenur Latif **Internal Examiner**
Assistant Professor
Department of CSE
Faculty of Science & Information Technology
Daffodil International University

Md. Abbas Ali Khan **Internal Examiner**
Senior Lecturer
Department of Computer Science and Engineering
Faculty of Science & Information Technology
Daffodil International University

Imran

Shah Md. Imran **External Examiner**
Industry Promotion Expert
LICT Project, ICT Division, Bangladesh

DECLARATION

We hereby declare that, this project has been done by us under the supervision of **Aniruddha Rakshit**, **Senior Lecturer**, **Department of CSE** Daffodil International University. We also declare that neither this project nor any part of this project has been submitted elsewhere for award of any degree or diploma.

Supervised by:

Aniruddha Rakshit

Aniruddha Rakshit
Senior Lecturer
Department of CSE
Daffodil International University

Co-Supervised by:

Md. Zahid Hasan
Assistant Professor
Department of CSE
Daffodil International University

Submitted by:

Md. Mehedi Hasan

Md. Mehedi Hasan
ID: 171-15-9021
Department of CSE
Daffodil International University

Susanta Debnath

Susanta Chandra Debnath
ID: 171-15-9093
Department of CSE
Daffodil International University

Mozahid

Md. Mozahid Hasan
ID: 171-15-9554
Department of CSE
Daffodil International University

ACKNOWLEDGEMENT

First, we express our heartiest thanks and gratefulness to almighty God for His divine blessing makes us possible to complete the final year project/internship successfully.

We really grateful and wish our profound our indebtedness to **Aniruddha Rakshit**, **Senior Lecturer**, Department of CSE Daffodil International University, Dhaka. Deep Knowledge & keen interest of our supervisor in the field of "*Machine Learning*" to carry out this project. His endless patience, scholarly guidance, continual encouragement, constant and energetic supervision, constructive criticism, valuable advice, reading many inferior drafts and correcting them at all stage have made it possible to complete this project.

We would like to express our heartiest gratitude to **Prof. Dr. Touhid Bhuiyan**, **Head**, Department of CSE, for his kind help to finish our project and also to other faculty member and the staff of CSE department of Daffodil International University.

We would like to thank our entire course mate in Daffodil International University, who took part in this discuss while completing the course work.

Finally, we must acknowledge with due respect the constant support and patients of our parents.

ABSTRACT

Technological breakthroughs and the interest of business entities have made the categorization of media products increasingly conventional in this digital environment. This is usually often a multilabel scenario in which an object might be labeled with several categories. Most of the literature addresses the movie genre classification as a mono-labeling task, generally based on audio-visual features. This study addressed a multilabel movie genre classification model using both supervised and unsupervised machine learning techniques to classify the movies into their corresponding genres. We created a dataset consisting of English subtitle files taken from The Movie Database (IMDB), which contains 1200 movies and each of the movies was labeled according to a set of eleven genre labels. We experimented with two feature extraction methods combined with the classifiers and a feature selection technique to reduce the dimensionality of our proposed work. In this study, we compared the performance of unsupervised and supervised techniques for the classification using several standard performance measures using both feature representation methods. We assessed that the best performers of the unsupervised techniques are K-means and Bisecting k-means in the term of cluster quality. In contrast, we observed the model evaluation using KNN, SVM and DT and find that SVM is better than the other classifiers among the supervised techniques. Finally, we compared the unsupervised and supervised technique in the term of quality of the clusters. We observed that the K-Means and Bisecting K-Means of unsupervised technique produced the cluster of higher quality than the SVM, DT and KNN supervised technique. We addressed the reason for the outliers of the training set and recommended to use unsupervised techniques to improve the assignment of predefining the categories and labeling the textual documents in the training set.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1
Introduction

1.1 Introduction

In movie theory, genres are several forms of recognizable types, categories, classifications with the same pattern, similar, or instantly identifiable. The different movie belongs to different kinds of the genre such as action, crime, drama mystery, thriller, etc. Before watching a movie, people usually find out genre information of the movie by their own interest as many of them love a specific genre or maybe more than one. On occasion, the actual genre to identify is quite difficult for viewers when they are watching movies. Categorizing or classifying movies makes it easier for the viewers to find out whatever they want. So, we need such a system that classifies movies according to their genre and people can conveniently recognize the genre information of the movies. The proposed method takes advantage of machine learning, Natural Language Processing, and raw text analysis of subtitle files.

1.2 Motivation

In this modern era, multimedia technology increases more streaming video providers, probably due to the consolidation of video on demand as a practical and convenient way aimed at allowing consumers to have access to movies, series, documentaries, etc. In the present time, various web portal host movies allow users to browse and watch online movies such as Netflix, Amazon Prime, etc. Simultaneously, the research community for multimedia retrieval has been devoting attempts to assess new techniques and methods that seek to adequately explore and retrieve movies based on data sources commonly available data sources with movie titles. In this respect, many studies focus on synopsis texts, trailer content (audio and/or image), posters images and so on [1, 2, 3].

1.3 Machine Learning

The most basic and important task in machine learning is classifying text or documents [4]. Text or document classification is intended for grouping text into single or multiple labels that are already predetermined [5]. The technique of text classification can be performed using supervised and unsupervised machine learning algorithms. Several supervised machine learning algorithms are used in text classification, including K Nearest Neighbor, Naive Bayes, Decision tree, Support Vector machine, etc. [6]. Clustering and association problems can be grouped as unsupervised machine learning. There are some common examples of unsupervised learning algorithms like K-Means clustering, Hierarchal clustering, Bisecting K-Means clustering, KNN (K Nearest Neighbors), etc. Nowadays, learning algorithms are more advance for classifying or categorizing text by the development of deep learning algorithms such as Recurrent Neural Network (RNN), Convolutional Neural Network (CNN) and Long Short-Term Memory (LSTM) [7].

1.4 Research Overview

In this study, we developed a multilabel movie genre classification model using several supervised and unsupervised machine learning algorithms to classify the movies into their corresponding genres. These techniques focus on the raw subtitle scripts of the movies. In our classification system, the supervised machine learning algorithms that we used are Support Vector Machine (SVM), Decision tree (DT) and K Nearest Neighbor (KNN). The unsupervised machine learning algorithms we used are K-means clustering, Bisecting K-Means Clustering and Agglomerative Hierarchical Clustering. We combined the algorithms with TF-IDF and Bag of Word (BOW) as the feature extraction method. To select the high ranked feature, Chi-square feature selection is used. In order to get the best model, each of the algorithms has evaluated using several metrics, mainly F1-score.

To solve the problems and perform genre identification accurately, some machine learning and deep learning techniques are implemented based on various data such as movie subtitles files, trailers, posters, audio-video content, etc. One approach is proposed in [8], where they proposed a smart trailer system that automatically makes a trailer through movie subtitle files. At the same time, the framework will extract the textual features to classify the familiar genre of the movie. However, their work is not efficient for classifying multiple movie genres as they showed only one genre in their proposed framework.

The main contributions of this work are:

- The approaches of machine learning technique to perform the multilabel genre classification.
- We constructed a dataset for raw subtitle-based multi-genre classification.
- The diversity of performance using different machine learning algorithms has been shown in our classification model.
- We optimized the algorithms using different parameters and reconfiguring their values.

The rest of the chapter is constructed as follows: In chapter 2, we present the related work review. In chapter 3, we propose a methodology of the article. In chapter 4, we describe the data set. In chapter 5, we evaluate the experiment and result and finally, in chapter 6, we conclude our work.

CHAPTER 2

Background Study

2.1 Introduction

The machine learning approaches is a way to develop an automated process. Defining the problem appropriately in the first step in building a machine learning model is data collection and goal defining. The machine learning process can help to classify the genre model perfectly. In this domain, many researchers have attempted to classify the movie genre through new approaches using machine learning and deep learning technique in the past. In this section, we have presented a couple of these works sequentially.

2.2 Related Works

Ertugrul A. M et al., have completed their research on movie genre classification from plot summaries using Bidirectional Long short time memory in their above model. Before watching a movie, people generally read plot summaries to get an idea about the movie genre. For this, they proposed a system to classify movie plot summaries according to their genre. Their approach separates each plot summary of a movie into sentences and tags each sentence according to its genre. They trained the model using Bi-LSTM networks for tagging the significant word and represent each word into sentences. To perform better results in the Bi-LSTM network, they compared the Recurrent Neural Network and Logistic regression model. Their above system can classify four types of genre: Thriller, Horror, Comedy, and Drama [9].

Mangolin R. B et al., in their research they addressed multilabel classification of the movie genre in various techniques. Their dataset takes input as several audio and images from movie trailers, subtitles, synopses, and movie posters from different movie scenarios. In their proposed system, they trained multilabel classifiers using Binary Relevance and ML-KNN. They used a large movie data set of 10,594 where each movie data set represents 18 genres and created multilabel classifiers using various representations. They assessed one and more individuals and combined the models trained with each other by late fusion. In their above system best result was obtained by the representation of LSTM [10].

Dharmadhikari S.C et al., have performed their research on various machine learning algorithms for text classification. Their research is working with three types of machine learning techniques respectively supervised, unsupervised, and semi-supervised. In every approach, they have shown the merits and demerits of every algorithm how it is efficient for various text classification tasks as scientific articles, news articles, spam filtering, identifying documents, genre, etc. For this proposed system, they have chosen the most common and effective algorithms in their work, respectively KNN, SVM, Decision Tree, etc., for supervised learning; at the same time, they choose unsupervised learning k-means and for semi-supervised learning, they choose the graph-based algorithm. Finally, they have shown that data source feature representation is the key factor of performing better in automatic text classification in machine learning approaches [11].

Shambharkar P. G et al., designed a 3D Convolutional neural network that classifies movie genres based on various movie trailers. Along with captures, the movie trailer provides temporary information and spatial features. Their proposed model designs eight layers in a 3D model and takes input as a 64 frames size of 224x224. Simultaneously, in 3D ConvNets, they applied different operations in each input location to get the temporary and spatial features. To get better performance, they trained both the VGG16 and Inception V3 2D CNN models on the same dataset used in the 3D model. Based on the input, the 3D neural network outputs one of the four genres: comedy, horror, action, and romance. And they obtained the best output in terms of accuracy 82.14%, in their 3D CNN Model [12].

Wang H et al., have presented a technique to classify the movie genre preference based on customer's natural feedback, social media information in their proposed system. This method is implemented through the Gaussian Kernel Support vector machine (SVM) to classify the model and logistic regression for extracting the features for sample data. They choose sample data randomly to be divided into five equal parts. Along with the for-selection process, they used cross-validation error in their model. They compared both the machine learning algorithm in terms of the error in sample and error out the sample and achieved the best accuracy in Gaussian Kernel Support vector machine 88% and 84%, respectively [13].

Wi J. A. et al., came up with an approach that used the Gram layer in a convolutional neural network (CNN) for classifying the multiple movie genres based on the movie poster. Using the Gram matrix, they generated a feature map of the movie poster image. They extracted the style features in the Gram layer using inter-channel relations. After that, they reproduced the movie poster data set for multilabel classification with up to 9 genres. Activation map helps to gram layer to focus on the style of the movie poster in their model. After that, they compared their model with the existing model [14].

Huang Y. F. et al., worked on a novel approach that uses the Self Adaptive Harmony Search (SAHS) algorithm in selecting features for various movie genres. Their methodology on the support vector machine algorithm is filled with features extracted from individual movie trailers, consisting of 277 features. Also, they applied the majority voting technique to estimate the genre of the movie in their model. Their methodology used both visual features along with the audio characteristic in this voting technique and use 25 features that differentiate of movie genre [15].

CHAPTER 3

Research Methodology

3.1 Introduction

This research paper followed a proper methodology to complete our research. In chapter 3, we are going to discuss the entire methodology of the research work. This section included a detailed discussion multilabel text classification of using the both supervised and unsupervised machine learning approach with a short explanation of every part of the methodology. The objectives of this thesis are to build a movie genre classification from movie subtitle model and a comparative analysis among classifier algorithms to choose the best out of them. In our research, we have used both supervised and unsupervised learning classifier techniques to get the best accuracy for this predicting model. First, we have collected a dataset that contains all of our necessary data for our research. Every machine learning model needs a good dataset to find an accurate automatic system. To complete the research primary and secondary source of data was collected and utilized.

3.2 Multilabel Text Classification

Classification is a standardized procedure of predetermined rules or standards determining whether objects belong in a given category [16]. A classification is intended to reduce individuals or groups effort to manage documents and even retrieve document information [17]. Multilabel classification and the multi-output classification seem to be the variants of the classification problem, where each instance may be assigned several labels. The multilabel classification was obtained from a text categorization problem investigation, where each document might become part of several predefined topics simultaneously. A multilabel classification assigns each sample to a set of target labels. This can be regarded as the prediction concerning properties of data points that are not mutually individual, such as a movie containing more than one or two genres at once. For example, 'Beyond the Mask' movie has action, adventure and drama genres at once. The multilabel classification involves a training set of instances, each corresponding to a set of labels. The task is to predict unseen instance label sets by analyzing training instances of known label sets. We defined multiple genres for each movie subtitle in our training dataset.

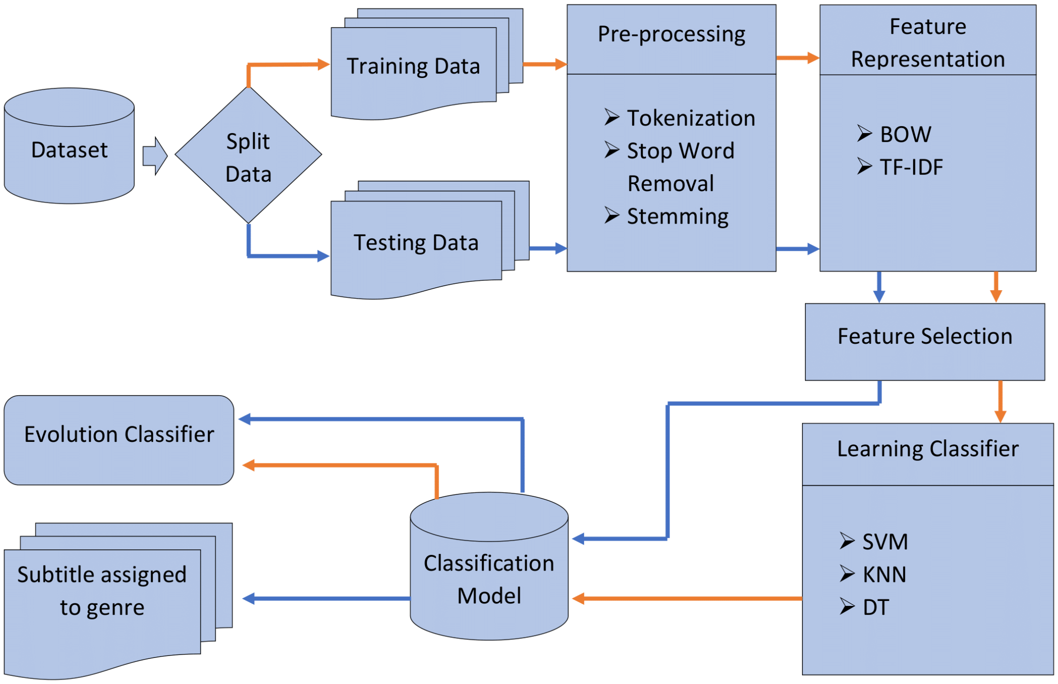


Figure 3.2.1: Movie genre classification model

3.3 Supervised Machine Learning

A standard formulation of the supervised learning task is the classification problem. Therefore, the learner should learn a function mapping a vector into one of the diverse classes by evaluating several examples of input-output of a function. The learning process in inductive machine learning aims to learn a set of rules from instances or build a classifier for the generalization of new instances. Figure 3.3.1 illustrates how supervised ML is applied to a real-world problem. This work mainly focuses on ML algorithms classification, also determining the most efficient algorithm with their highest accuracy and precision. Besides assessing the performance of different algorithms on small and large data sets with a view classify them correctly and provide insight into how supervised machine learning models can be built.

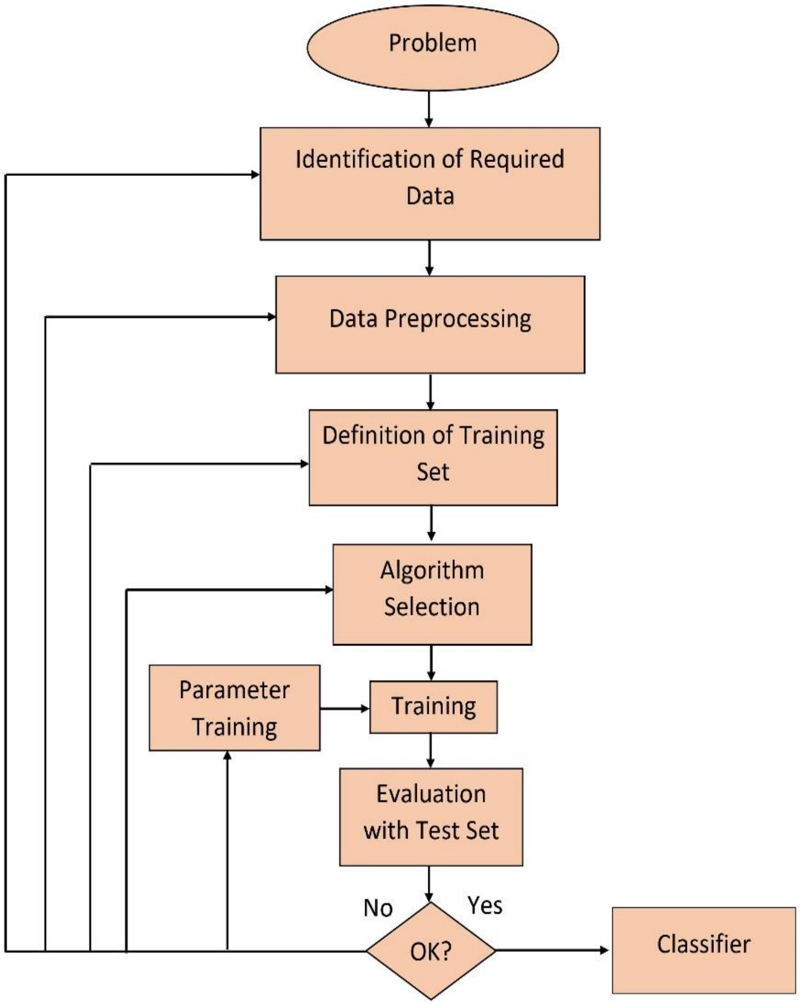


Figure 3.3.1: Supervised machine learning process diagram

3.3.1 Support Vector Machine

Support Vector Machine has a robust model capable of handling various tasks, including linear or nonlinear classification, regression, and outward detection. SVM trains to create maximum margin at the decision-making threshold, which creates the position between two classes by few instances of train data. These data points are known as support vectors. By calculating the distance between support vectors and the new instance, SVM classifies the new instance. The training set of instance-label pairs is given as $(\mathcal{X}_{i,} Y_i), i = 1, \ldots., l$ where $\mathcal{X}_i \in R^n$ and $Y \in \{1, -1\}^i$. The following optimization problem must be accomplished by SVM [19].

$$\min_{w,b,\xi} \frac{1}{2} W^T W + C \sum_{i=1}^{1} \xi_i \tag{1}$$

$$\text{Subject to } y^i (W^T \phi(X_i) + b) \geq 1 - \xi_i, \xi_i \geq 0 \tag{2}$$

Were, function ϕ map training vector X_i into a higher dimension space. $K(X_i,\ X_j) \equiv \phi(X_i)^T \phi(X_j)$ is known as the kernel function. There are many kernel functions such as polynomial, linear, sigmoid, and RBF (Radial Basis Function). We used the most prevalent RBF kernel, and the equation is following.

$$K(X_i, X_j) \equiv \exp(-\gamma\ \| X_i - X_j \|^2), \gamma > 0 \tag{3}$$

SVM is the responsive classifier for scaling the input data. Kernel values in SVM are usually dependent on the internal components of feature vectors. Hence, large scale values also can cause numeric problems. Hsu recommends the range [-1, +1] or [0, 1] is linearly scaled for each attribute [18]. We optimized the SVM classifier in our model to obtain better performance using different value of C, where C evaluates the impact of the misclassification on the objective function. When the value of C is 1, we got the highest accuracy in our model. We also used linear kernel as there is a large number of features we have in our dataset.

3.3.2 Decision Tree

A Decision Tree is a tree model that is frequently used for regression and classification. It learns simple decision rules that separate the node from features to predict the target value. The CART (Classification and Regression Tree) algorithm is used for training Decision Tree that divides the node into two sub-sets through the single feature k and its threshold t_k [19]. The cost function that the CART algorithm aims to minimize is the following equation (4).

$$J(k, t_k) = \frac{m_{left}}{m} G_{left} + \frac{m_{right}}{m} G_{right} \tag{4}$$

Where, $m_{left/right}$ has denoted as the number of instances in the left and right subset and $G_{left/right}$ has denoted as the impurity of left and right subset. Different functional methods are available to measure impurity such as Gini and cross-entropy. We defined the range of maximum depth using hyper parameter tuning method in Decision Tree. We set the range using different values. When value of maximum depth is 100000 and the random state value is 0, the model performs best.

3.3.3 K-Nearest Neighbor

The K Nearest Neighbor is one of the simple and most fundamental classification techniques that estimates the significant portion of the k nearest neighbors in a feature space belong to a certain category [20]. The algorithm includes several key factors: k-value selection, distance measurement and so on [20]. Initially, an experiment is carried out for the selection of the k value and the optimal k value is selected through a simple cross validation process from the text data. The test set would also be validated by using a training set model to interfere with the training set and the test set sample selection. We got the best performance when the value of k is 4. Moreover, distance measurements are used to measure the distance between individuals in a space. Euclidean distance is the most used metric for the measurement of the absolute distance between points in a multidimensional space.

$$Dist(X, Y) = \sqrt{\sum_{i=1}^{n} (x_i - y_i)^2} \tag{5}$$

Minkowski distance is the common measurement that indicates the distance between numerical points and it’s not only a distance but also a set of definitions of distance. Between two n-dimensional variables, the Minkowski distance A= (x11, x12,..., x1n) and B= (x21, x22,….., x2n) is defined as:

$$D_{12} = \sqrt[p]{\sum_{k=1}^{n} |x_{1k} - x_{2k}|^{p}} \tag{6}$$

Our experimental result for the KNN algorithm is shown in table 5.3.1 for different values of k. When k = 4, the model performs best and delivers the higher accuracy.

3.4 Unsupervised Machine Learning

Unsupervised Learning is a technique of machine learning where the models of training datasets are not supervised. Rather, the hidden patterns and insight are retrieved from data using the models. It can be related to the learning of the human brain while new things are being learned. In our research, supervised learning aims to identify the underlying structure of our movie subtitle dataset, represent them in a compressed format, and group them according to the similarities.

3.4.1 K-Means Clustering

The principle of the K-Means algorithm is that the mean of the documents allocated to the cluster can be represented by each of the k clusters assumed to be the centroid of that cluster, discussed by Hartigan [21]. In our experiments, we used the second version of k-means algorithm is known as the incremental or online version. It is discussed by Berkhin [22] and Steinbach et al. [23] that in the domain of text document collections, online k-means performs better than the batch version. Initially, k documents are selected randomly from the corpus as the initial centroids. Then, the documents are assigned iteratively and after each of the assignment of a document to its nearest centroid, centroids are updated incrementally. When no reassignments of documents occur, the iteration stops. The centroid vector c of cluster C of documents has defined as follows:

$$c = \frac{\sum d \epsilon C^{d}}{|C|} \tag{7}$$

Where c is determined by the average weight of terms of the documents in C. The similarity between a centroid vector c and a document d has defined by cosine similarity measure as:

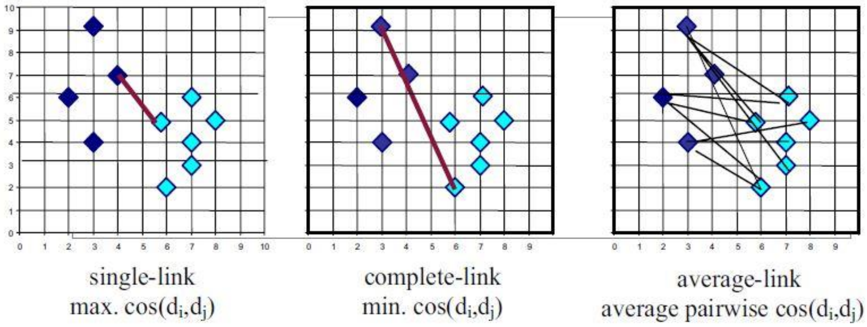
$$cos(d, c) = \frac{d \bullet c}{||d||\ ||c||} \tag{8}$$

3.4.2 Bisecting K-Means Clustering

Bisecting K-Means is a clustering algorithm that achieves a cluster of hierarchy through the repeated application of the basic k-means algorithm. In each step, a cluster is selected to be split by applying basic k-means for k = 3 in Bisecting K-Means. The clusters with the least overall similarity or the cluster with the maximum number of documents may be assigned to separate can be chosen to be split. In both cases, we conducted experiments and observed the similar performance. Therefore, only when the largest cluster is chosen for split, we reveal the result.

3.4.3 Agglomerative Hierarchical Clustering

Agglomerative clustering algorithms starts in a separate cluster with each document and combine the most related clusters at each iteration until the stop criteria is met. They are primarily categorized as single ties, complete links and decent links and they define their inter-cluster similarity depending on this categorization. The principle is clarified in figure 3.4.3.1.



single-link
max. $\cos(d_i,d_j)$

complete-link
min. $\cos(d_i,d_j)$

average-link
average pairwise $\cos(d_i,d_j)$

Figure 3.4.3.1: Inter cluster similarity

- The single link technique determines the resemblance of two clusters C_i and C_j represents the similarity of two documents $d_i \in C_i$ and $d_j \in C_j$ which are most similar.

$$Similarity\ \mathcal{Y}_{single-link}(C_i, C_j) = \max_{d_i \in C_i, d_j \in C_j} |cos(d_{i,} d_j)| \qquad (9)$$

- The complete link technique determines the similarity of two clusters C_i and C_j represents the similarity of the two documents $d_i \in C_i$ and $d_j \in C_j$ which are least similar.

$$Similarity\ \mathcal{Y}_{complete-link}(C_i, C_j) = \min_{d_i \in C_i, d_j \in C_j} |cos(d_{i,} d_j)| \qquad (10)$$

- The average link technique determines the resemblance of two clusters C_i and C_j as the pairwise average similarities of the documents from each cluster where n_i and n_j are sizes of clusters C_i and C_j respectively.

$$Similarity\ \mathcal{Y}_{average-link}(C_i, C_j) = \frac{\sum d_i \in C_i, d_j \in C_j\ |cos(d_{i,} d_j)|}{n_i n_j} \qquad (11)$$

CHAPTER 4
Experimental Settings

4.1 Introduction

The movie subtitle classification generally consists of three primary stages, the preprocessing stage, the feature extraction stage, and finally, the document classification stage. The method that turns the text into an appropriate format is included in the preprocessing stage. The feature extraction stage is the process whereby the features are extracted and converted into a numeric vector. Finally, the classification stage of the document including the evaluation and construction of the classification model.

4.2 Dataset Characteristics

The model includes dataset as the set of collection about 1200 movie subtitle files which has distributed in 11 main classes such as action, drama, comedy, adventure, crime, thriller, horror, mystery, romance, family, and fantasy that has taken from the YIFY Subtitle website [24]. Each of the movie genres consists of approximately 100 movies and their subtitles (fig 4.2.1). The subtitle file collection has done by choosing only the English language, based on the most popular movies on IMDB. We downloaded the subtitle file as .srt format and converted it into .txt format. We defined multiple genres for each of the movie subtitles in our dataset. Our movie subtitle dataset to be split into two models: training model and testing model. The former refers to the pre-classified set of data used to train the classifier. On the other hand, training models are manually identified to support the classifiers of different movie subtitle to create libraries for each of the movie subtitles, while the accuracy of the classifier has determined in the test model by counting the correct and incorrect classifications for each movie subtitle in the set classified into suitable main classes by the classifier.

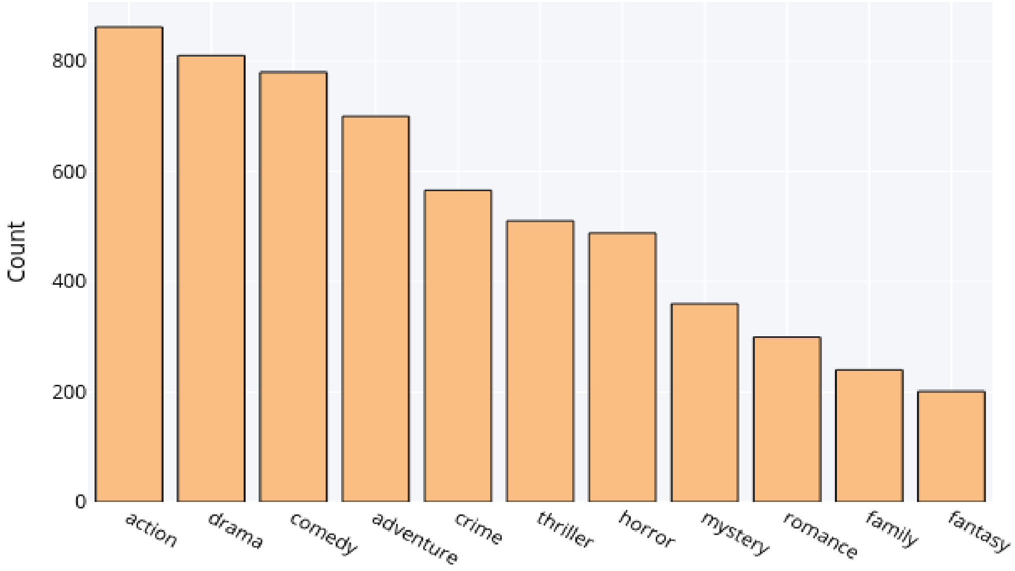


Figure 4.2.1: Total number of movies for each genre

4.3 Data Preprocessing

Our aim behind data preprocessing is to represent the dataset as a feature vector to split the text into individual words. After reading the input text documents, the text preprocessing step separates the text into features named tokenization, words, terms, or attributes. This text document is represented as a vector space in a data representation whose components consist of features and their weights obtained by the frequency of each feature in that text document. It then erases non-informational characteristics, including special characters, numbers, and stop words. The rest of the features are further streamlined through stemming process by reducing them to their root.

4.3.1 Tokenization

The first step is to normalize the texts by removing tags from HTML and other tags. Tokenization is the process whereby a text stream is broken into words, phrases, symbols or other meaningful elements known as tokens. The aim of the tokenization is to explore the words in a sentence for our movie subtitle dataset. For further processing, the list of tokens becomes input, such as parsing or text classification. We used a regular expression-based tokenizer in order to break our text into useful tokens developed by Christopher Potts [25] and improved upon by Jganadg Gopinadhan [26].

4.3.2 Remove Stopwords

The stopwords are very generic words with little semantical relevance to the text, which are included in a pre-defined list known as Stopword lists. Their appearance in textual data represents an impediment to interpreting the documents' content due to their high frequencies. Stop words are the most frequently used words such as "and," "are," "this" etc. They are not efficient on the purposes of text classification. We used the NLTK stopwords list [27] to remove all stopwords from the subtitle in this work. In [38, 29], the impact of stopwords for text and music mood classification has been studied by the authors.

4.3.3 Stemming

Stemming is the approach where affixes are excluded from features, i.e., the process whereby the inflected words are reduced to their stem. For example, given the words “presentation,” “presented,” “presenting,” the result of a stemming algorithm could be “present” [30]. In [31], it became apparent that this technique is primarily used for reducing dimensionality in the field of text classification. It is also a quick procedure. It should be emphasized that the stemming algorithms are language-specific and therefore, for some languages it might not be available [32].

go back
new york
next day
mi ju
come back
metal gear
year old
young woman
next morning
return home
fox mask
yoon jin
go back
next day
make way
young connor
mr deere
master chief
come back
dr evil
al saleem
walk away
say will
dr raymond
pee wee
next morning
next day
miss madrigal
go back
big boy
walk away
new york
dr edwardes
return home
next day
next morning
pai mei
go back
kung fu
year old
say will
run away
new york
next day
next morning
adãlie
spider man
young woman
year later
eloã sa
new york
next day
next morning
go back
tall man
two men
year old
next day
miss honey
mr coulter
sam gold
young woman
run away
pu yi
new york
next morning
next day
year old
go back
seok woo
rock rock
want rock
give away
talkin bout
human human
give love

Figure 4.3.3.1: Representation of word stemming

4.3.4 N-Gram

N-Gram is the technique of generating the sequence of given data that split into chunks of size N. For example, given the term “next”: for n=2, the representative attributes would be “_n”, “ne”, “ex”, “xt” and “t_”; for n=3, “_ne”, “nex”, “ext” and “xt_”; and for n=4, “_nex”, “next” and “ext_” (“_” denotes the beginning and ending of the word) [33]. In the operational code sequence, Mikhail used the N-gram model to extract the essential feature and created an N-gram vector frequency [34]. This technique is generally used in distinct language cases when language have a common root. In this manner, the variability of some words does not impair the common morphemes between languages. Therefore, this method facilitates inducing morphemes that separate the words into character sequence of fixed size n, is predefined according to the context of the application. For text categorization tasks, Willeam used N-gram and got high accuracy 99.8% [35].

4.3.5 Data Representation

In the training set, each text (object) character is usually represented by a vector in shape (x, d) where x is equivalent to R^n, is a measurement vector and d is the class label. Each aspect of this space comprises a single feature and the weights of that vector, determined by each feature of the movie subtitle dataset by the frequency of occurrence. This study demonstrated each vector of dataset d as d (W_1, W_2..., W_n) where W_i is the weight of ith term of dataset d and this representation is called as vector space model or data representation. In this step, an initial weight for each feature is given and this weight may increase based on the input text dataset frequency of each feature.

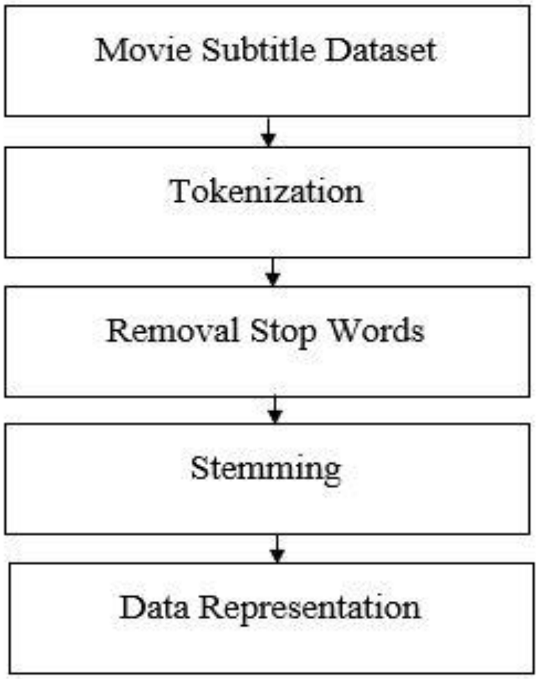
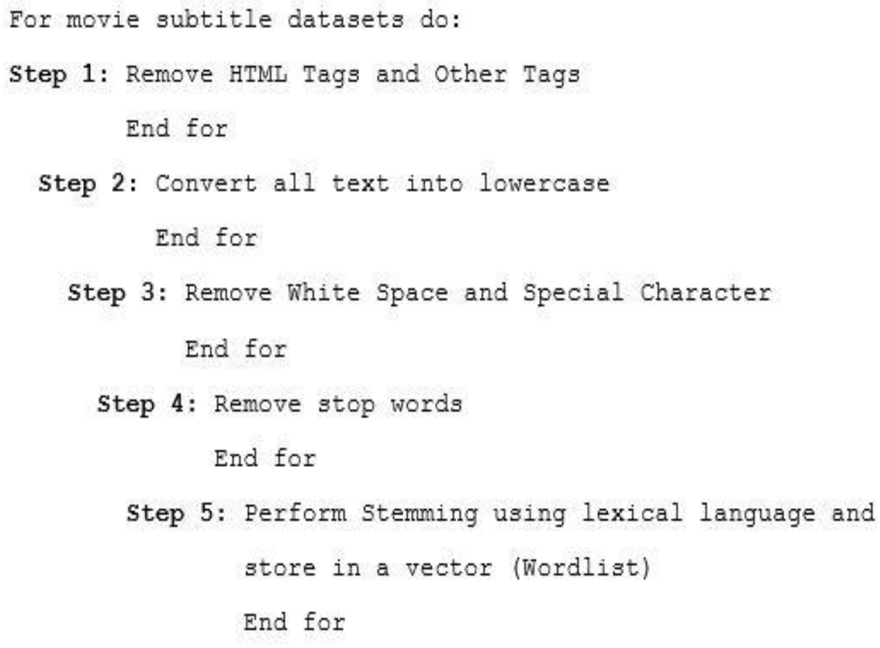


Figure 4.3.5.1: Text preprocessing steps

The weight of a feature is attributed to the sum of the initial conflated frequencies. These steps are used to prepare the text document mentioned above as seen in figure 4.3.5.1. The demonstration of pseudocode for which is used for data preprocessing is given below.



```
For movie subtitle datasets do:
Step 1: Remove HTML Tags and Other Tags
        End for
  Step 2: Convert all text into lowercase
          End for
    Step 3: Remove White Space and Special Character
            End for
      Step 4: Remove stop words
              End for
        Step 5: Perform Stemming using lexical language and
                store in a vector (Wordlist)
                End for
```

Figure 4.3.5.2: Data preprocessing pseudocode

4.4 Feature Extraction

In our research, we mentioned earlier that we will treat this multilabel classification problem as a Binary Relevance problem. Hence we encoded the target variable by using sklearn’s MultiLabelBinarizer(). We extracted features from our dataset using the Bag of Words (BOW) and TF-IDF model. In a bag of words approach (Bow), the order of terms throughout documents is omitted, and their frequencies of occurrence are used [36]. Within a text collection, each unique word is considered as a different feature. Consequently, a multi-dimensional feature vector represents a document. Each dimension corresponds to a value that is weighted by Term Frequency (TF) and Term Frequency-Inverse Document Frequency (TF-IDF) [37] in a feature vector. In particular, a collection of documents D holding N documents such that $D = \{d_0, \ldots \ldots, d_{n-1}\}$ each document d_i includes a group of terms t represented as vectors in the Vector Space Model (VSM) as follows:

$$d_{ij} = (t_{1j}., t_{2j}., \ldots \ldots \ldots \ldots., t_{ij}), j = 1, \ldots \ldots.., y \tag{12}$$

In equation 10, y determines the number of distinct words in the document. In order to calculate the weight of a word in a document, the TF-IDF technique applies the TF and DF by used Equations (13) and (14).

$$IDF(t) = \log(\frac{N}{DF(d,t)}) \tag{13}$$

$$TF - IDF = TF(t,d)IDF(d,t) \tag{14}$$

Where N denotes the entire documents in the training set, DF(d,t) refers to the number of documents comprising term t and TF(d, t) denotes the total number of times term t occurs in document d.

4.5 Feature Selection

Feature selection technique is the process that eliminates the irrelevant features and chooses the most relevant ones. The feature selection technique has been classified into three categories, namely wrappers, filters and embedded methods. Among these three, the Filter method is computationally fast. However, feature dependencies are not usually considered. In text classification domain, Filter-based methods [38] are used widely. The preference for distinguishing features in text classification is based on large numbers of filter-based techniques. In this study, Chi-square testing (x^2) is suggested, which is defined as a well-known method of testing discrete data hypothesis from statistics [39]. This method examines the correlation between two distinct variables and determines the independence or the correlation among them [40]. Equations 15 and 16 define the value (x^2) for each term t in a category c.

$$\chi^2(t_k, c_i) = \frac{|T_r|[p(t_k,\ c_i) * p\left(\overline{t_k}, \overline{c_i}\right) - p\left(t_k, \overline{c_i}\right) * p\left(\overline{t_k}, c_i\right)]}{p(t_k) * p\left(\overline{t_k}\right) * p(\overline{c_i})} \tag{15}$$

In addition, it is evaluated using:

$$\chi^2(t, c) = \frac{N * (AD - CB)^2}{(A + C) * (B + D) * (A + B) * (C + D)} \tag{16}$$

In case, A denotes the number of documents in category c, B refers to the number of documents not in category c and both are including term t. Gradually, C refers to the number documents in category c, D refers to the number of documents of a different category and both are not comprising term t. N refers to the entire documents.

4.6 Frequency Analysis

The frequency of attributes generated by representation for multilabel movie genre classification subtitle dataset using stopwords or after of stopwords removal are shown in figure 4.6.1. This representation has made with the frequency of top 20 attributes for unigram, bigram and trigram representation. The analysis of diverse representation shows that, as the parameter n increases for the n-gram representation, the number of attributes gradually decreases. The stemming representation includes more features that are highly dimensional compared to the bigram and trigram representation. Furthermore, the optional step in eliminating stopwords has little impact on the movie subtitle filtering out the common words.

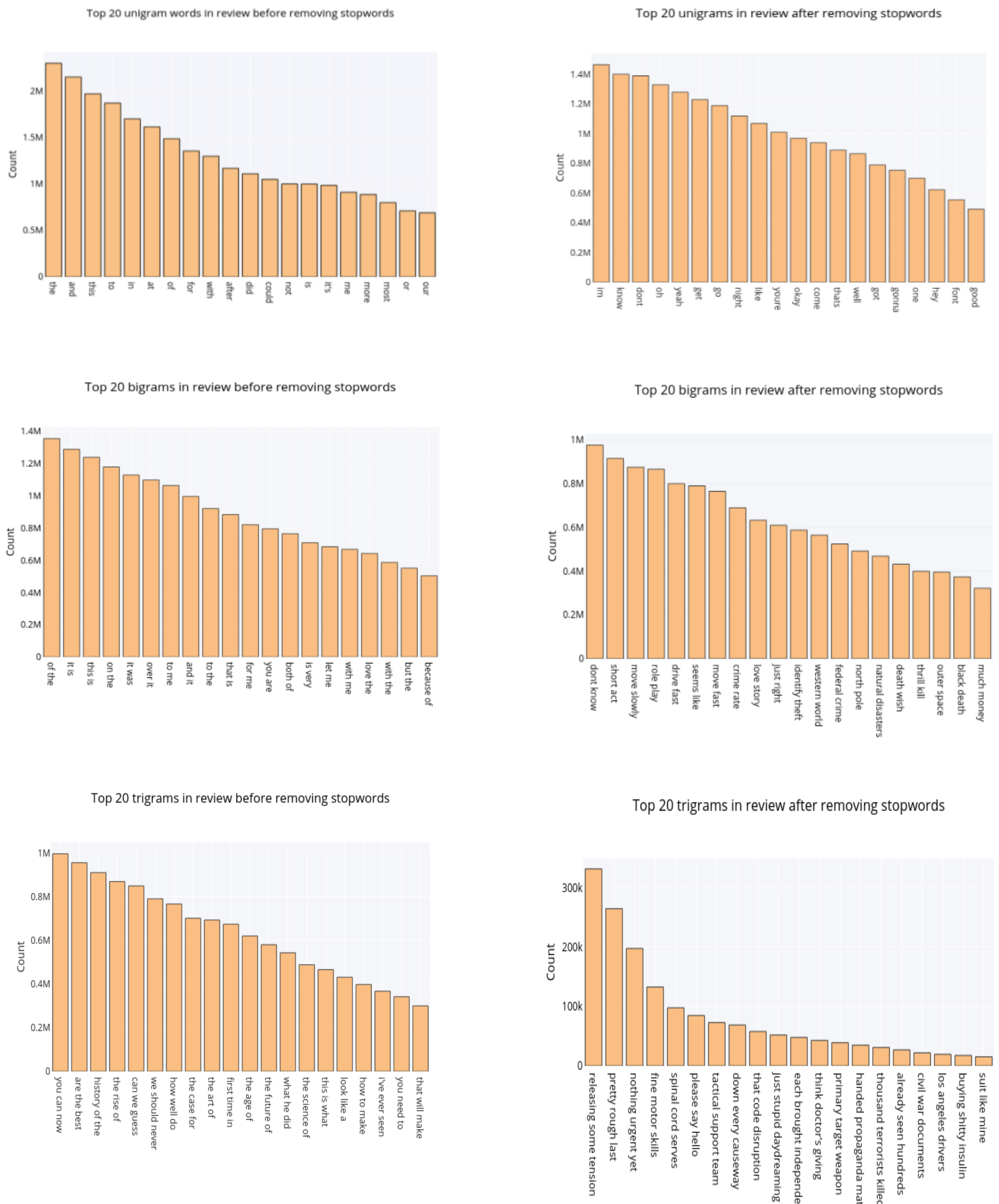


Figure 4.6.1: The frequency of top 20 attributes in N-Gram representation

CHAPTER 5
Model Evaluation

5.1 Introduction

In this study, we performed multi-label movie genre classification from movie subtitles using supervised machine learning approach where the class labels are action, drama, comedy, crime, thriller, horror and romance. Experimental tests and evaluations are reviewed to assess the validity of the proposed framework. The evaluations of the framework generally focus on the accuracy of classifying movie subtitles to their corresponding genre.

5.2 Evaluation Metrics

In these experiments, all subtitles in the dataset were prepared by converting them into UTF 8 encoding. All classification experiments that separated data into ten mutually exclusive subsets called folds were carried out in cross-validation. Each of the folds contains almost 120 movie subtitles. One of the subsets is used as a test set and all the reminder were used as the training set. Several mathematics rules such as precision (P), recall (R) and F-measure (F1-Score) is used to evaluate the performance for classification model of supervised learning to classify the movie subtitles into the correct genre are investigated as follows:

$$Precision = \frac{TP}{TP + FP} \tag{17}$$

$$Recall = \frac{TP}{TP + FN} \tag{18}$$

Where TP is the number of documents that appropriately assigned to the genre. TN implies the number of documents appropriately attributed to the negative group. FP denotes the number of documents a system wrongly attributed to the class. The number of documents is symbolized by FN, which falls into the class but are not assigned to the genre. For the study, micro-F1 score (a well-known F1 measure) [41] is chosen as the success measurement and could be calculated as follows:

$$Micro - F1 = \frac{2 * Precision * Recall}{Precision + Recall} \tag{19}$$

In unsupervised machine learning, we used two types of measurement to evaluate the cluster quality, measurement of internal quantity and measurement of external quality [25]. The quantification of internal quality does not use external resources, including information about the class label, to evaluate the produced clustering solution. On the opposite, the measurement of external quality relies on the labelled test document corpora. We used purity as an external quality measure, overall similarity as an internal quality measure, and two widely used external quality measures in text mining: entropy and F-measure [25] to evaluate the quality of the unsupervised clustering algorithms.

Overall consistency is a metric of internal quality that uses the weighted similarities of internal cluster similarities to measure the consistency of the clusters produced. The Internal cluster similarities I for cluster C_j can be defined as:

$$I_j = \frac{1}{n^2} \sum_{d \in C_j, d' \in C_j} \cos(d, d') \tag{20}$$

$$Overall\ Similarity\ = \sum_j \frac{n_j}{N} I_j \tag{21}$$

Where, n_j denotes the number of documents in cluster j and N refers to the entire documents in the corpus.

Purity measures the dimension of each cluster contains documents from one class. Purity for a particular cluster j of size n_j is defined to be:

$$p_j = \frac{1}{n_j} \max_i n_{ji} \tag{22}$$

Where, n_j denotes the number of documents of class i which are assigned to cluster j. Therefore, P represents the fraction of the aggregated cluster size is assigned to the cluster by the largest class of documents. The absolute purity of the clustering solution is accomplished by the weighted sum of each cluster purity.

$$P = \sum_j \frac{n_j}{N} P_j \tag{23}$$

Where N denotes the total number of documents among the collection of documents. In general, the higher the purity values, the better is the clustering solution.

Entropy measures the homogeneity of the clusters. The optimal solution for clustering leads to clusters consisting of documents from only one class. The entropy is zero in this case. Generally, the clusters are more homogenous when the entropy is lower. The overall entropy E is the sum of the entropies E of each cluster J for a set of clusters.

$$E_j = -\sum_i P(i,j).logP(i,j) \tag{24}$$

$$E = \sum_j \frac{n_j}{N} E_j \tag{25}$$

P(i, j) denotes the probability of a document with the class label i which also assigned to j, n_j refers to the size of cluster j and N represents the entire documents in the corpus.

5.3 Results and Discussions

In this study, we performed multilabel movie genre classification from movie subtitles using supervised and unsupervised machine learning approach where the class labels are action, drama, comedy, adventure, crime, thriller, horror, mystery, romance, family, and fantasy. Experimental tests and evaluations are reviewed to assess the validity of the proposed framework. The evaluations of the framework generally focus on the accuracy of classifying movie subtitles to their corresponding genre.

Figure 5.3.1 shows the F1-score by genre for both BOW and TF-IDF feature extraction methods. It can be seen that the model for the TF-IDF feature extraction method consistently attains a better F1-score than the model that uses the BOW feature extraction method. The KNN+TF-IDF model is competitive with the KNN+BOW model on the most popular genres such as 0.78 to action, 0.76 to crime, 0.75 to thriller, 0.73 to comedy, 0.72 to adventure and 0.71 to mystery but under-performs on less popular genres. The romance class was mostly misclassified by the DT+BOW and DT+TF-IDF, 0.79 to action and 0.77 to crime for the DT+BOW and 0.81 both to action and crime for the DT+TF-IDF. The drama class has mostly been misclassified by SVM+BOW and SVM+TF-IDF, but stronger than the DT+BOW and DT+TF-IDF. The F1-score for the SVM+TF-IDF is better than SVM+BOW for all of the genres. In most of the cases drama and romance class is mostly misclassified. After examining more deeper into the dataset, this is owing to the similarities of movie subtitles between the drama and romance class.

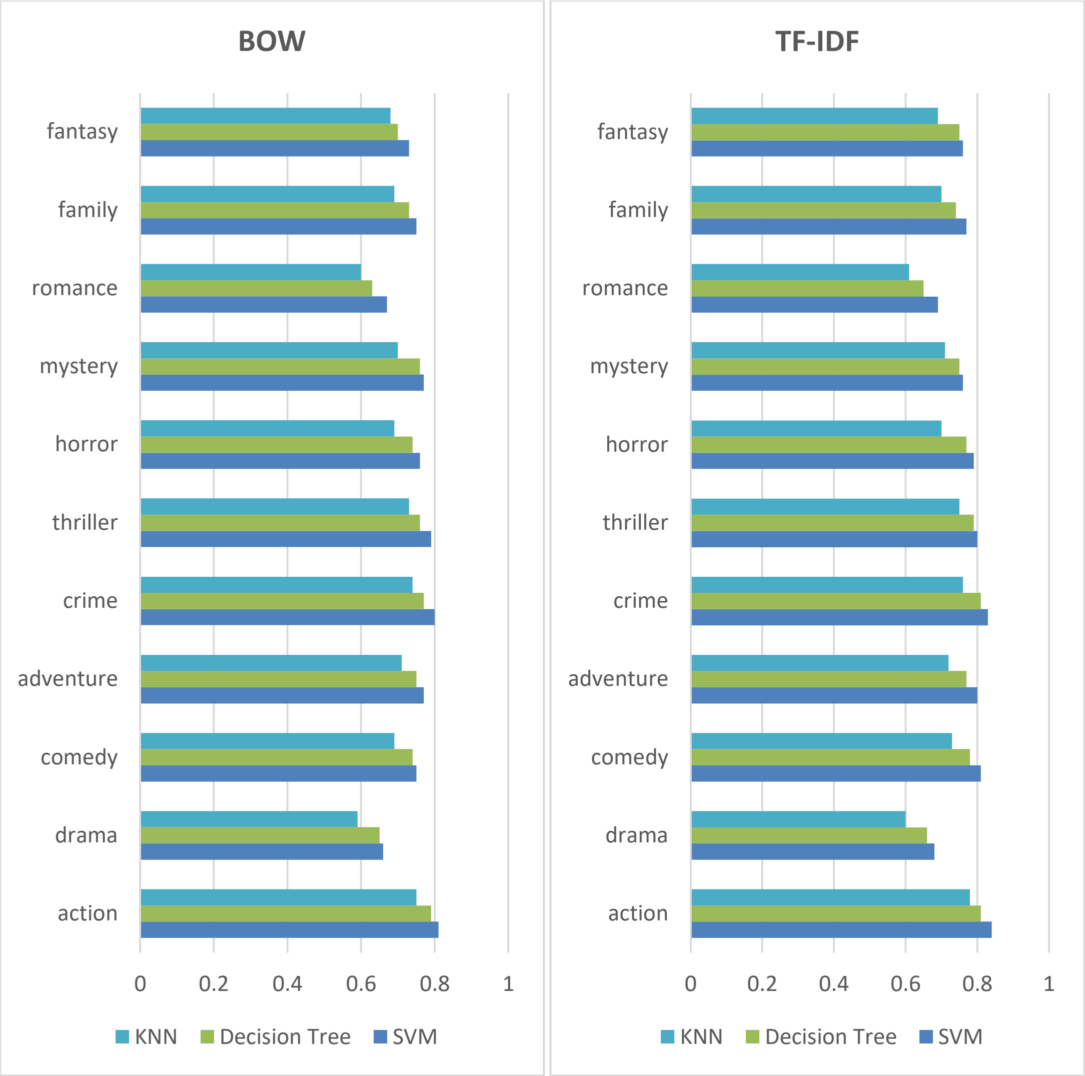


Figure 5.3.1: F1-score by genre

The accuracy obtained from supervised machine learning classifiers using TF-IDF and Bag of Word (BOW) representation on the movie subtitle dataset is listed in table 5.3.1. Based on the applied methods, namely BOW and TF-IDF feature representation approaches, SVM, DT, KNN classifiers, Chi-square feature selection method and applying of different values of the parameter has a different impact on the accuracy of all the classifiers in movie subtitle data classification. On BOW feature representation, the SVM classification approach was the most robust classifier amongst all the classifiers, which has low dependency on implementing different values of the parameter. It turned out that SVM, followed by KNN and DT achieved the highest F1-measurement, while the measurement is achieved greater by DT than KNN. In TF-IDF feature representation, the SVM classifier can always achieve high classification accuracy with the low diversity of classification accuracy across the tested value of parameters. The DT classifier were found to be the strongest classifier rather than KNN. In the case of DT, it achieved better performance than the Binary BOW feature vector but still had a high variance. Moreover, we have noticed that KNN got the lowest F1-measurement among all the classifiers for both BOW and TF-IDF representation methods. The alleged cause of the low performance of the KNN method due to the difficulties to evaluate the optimal value of K. The total value of K is important because the findings of the KNN probability are determined from the samples of K.

TABLE 5.3.1: PERFORMANCE COMPARISON USING BOTH FEATURE REPRESENTATION METHOD

Algorithm	Metrics	Feature Extraction	
		BOW	TF-IDF
SVM	Precision	0.86	0.90
	Recall	0.83	0.87
	F1 Score	0.91	0.93
DT	Precision	0.84	0.88
	Recall	0.80	0.83
	F1 Score	0.86	0.87
KNN	Precision	0.80	0.85
	Recall	0.79	0.83
	F1 Score	0.81	0.84

Several other things can influence the performance of classifiers, such as the dimension of input data. We did the second additional experiment to demonstrate output variations in each feature vector according to the data dimension. In the movie subtitle classification domain, N-gram is effective because subtitle has a sequence to classify them according to the genre. In table 5.3.2, the measurement is shown for bigram and trigram representation with both TF-IDF and Bag of Word (BOW) feature extraction methods. In trigram with BOW feature vector, the F1- measurement for KNN is the same as bigram, but for SVM and DT, the performance decreases in trigram rather than bigram. In trigram with the TF-IDF feature extraction method, the performance for both SVM and KNN has increased, DT remains the same.

TABLE 5.3.2: PERFORMANCE COMPARISON OF EACH CLASSIFIER COMBINING FEATURES WITH N-GRAM

Algorithm	Metrics	Feature Extraction			
		BOW+BIGRAM	TFIDF+BIGRAM	BOW+ TRIGRAM	TFIDF+TIGRAM
SVM	Precision	0.84	0.90	0.83	0.92
	Recall	0.86	0.87	0.85	0.90
	F1 Score	0.86	0.93	0.85	0.94
DT	Precision	0.82	0.87	0.84	0.86
	Recall	0.80	0.84	0.83	0.85
	F1 Score	0.84	0.88	0.82	0.88
KNN	Precision	0.82	0.84	0.83	0.84
	Recall	0.80	0.82	0.79	0.82
	F1 Score	0.82	0.85	0.82	0.86

Figure 5.3.2 displays the performance of k-means, Bisecting K-Means; single link, average link and complete link agglomerative hierarchical algorithms in terms of entropy, purity, F-measure and overall similarity evaluation metrics using TF-IDF and BOW feature representation method over movie subtitle dataset. Among the agglomerative hierarchical clustering algorithms, the single-link algorithm performs considerably worse than the other algorithms for both feature representation methods. Each document is assigned to the cluster of its nearest neighbor by this algorithm. Yet, any two documents may share several of the same terms and be nearest neighbors without belonging to the same class. In our movie subtitle dataset, a large number of documents are nearest neighbors belong to different topics. These properties make our movie subtitle dataset less complicated. In the TF-IDF representation method, the performance of the average link is a little bit improved rather than the BOW representation method. For both of these methods, the average-link performs best out of agglomerative hierarchical clustering algorithms. We intuitively explained the reasons for the poor performance of the single link. The complete link algorithm is built on the presumption that all the documents in the cluster are very similar. The high dimensional diversity of the text document domain does not take this assumption into consideration, where each distinct word is considered as a different feature and context knowledge such as hypernyms, hyponyms and synonyms are not considered. By relying on more global properties, the average-like algorithm overcomes these problems in measuring cluster similarity. The similarities of the two clusters are evaluated by taking all the documents in both clusters into consideration in this algorithm. When purity, entropy and overall similarity metrics are considered, the online k-means and Bisecting K-Means perform better than single link and complete link agglomerative hierarchical clustering algorithms for both TF-IDF and BOW representation method. Their performance is either similar or better to the average-link algorithm. With respect to F-measure, Bisecting K-Means performs better than k-means on TF-IDF representation. In the case of BOW representation, the performance of k-means is better than Bisecting K-Means and in both cases, Bisecting K-Means is better than average-link clustering algorithms.

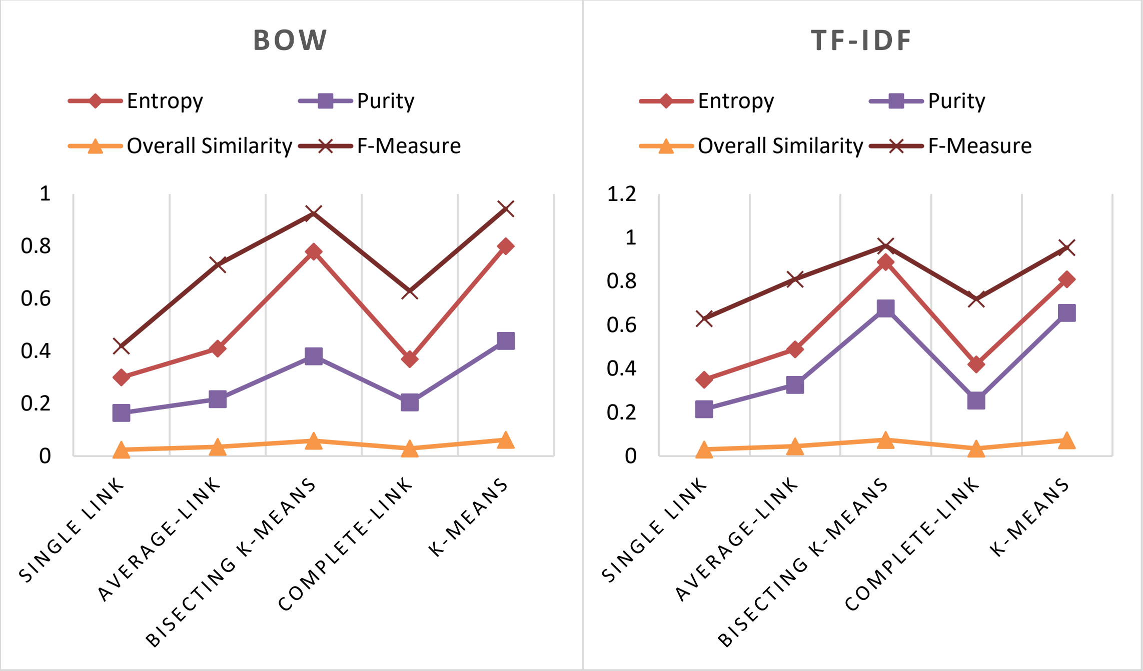


Figure 5.3.2: Performance measurement of unsupervised learning method

In this literature, the comparison of the supervised [42] and unsupervised [43] technique has been performed. The quality of cluster produced by both approaches has been compared. The classification solutions obtained through the supervised techniques are known as cluster solutions and are assessed through the evaluation metrics used to evaluate the performance of clustering algorithms. The clustering algorithm for each document corpus is the number of clusters equivalent to the number of predefined classes. Figure 5.3.3 display the results for our movie subtitle dataset respectively. The difference from the unsupervised technique is that, to resemblance information between documents, the supervised techniques employ class label information. Therefore, the clusters (groups) obtained by Supervised techniques are expected to be of higher quality than the Unsupervised techniques. However, we can observe that the best performers of the unsupervised techniques are K-Means and Bisecting K-Means, which generally achieve better performance than SVM, DT and KNN, in terms of entropy, purity, overall similarity and F-measure in TF-IDF feature representation method. Bisecting K-Means achieves the highest performance in terms of F-measure. On the other hand, the performance of Bisecting K-Means is worse and K-Means performs best from all other supervised algorithms in BOW representation in terms of entropy, purity, overall similarity and F-measure. Another observation is that unsupervised techniques generally yield better overall similarity efficiency than supervised techniques because they make a decision only based on the information of similarity between documents. On the contrary, the techniques of supervised learning use a labeled training set. This remark made us think that some outliers in the labeled training set may lead to a decrease in overall similarities between the obtained clusters and unsupervised techniques could be applied to enhance the task in the training set of predefined categories and labeling documents.

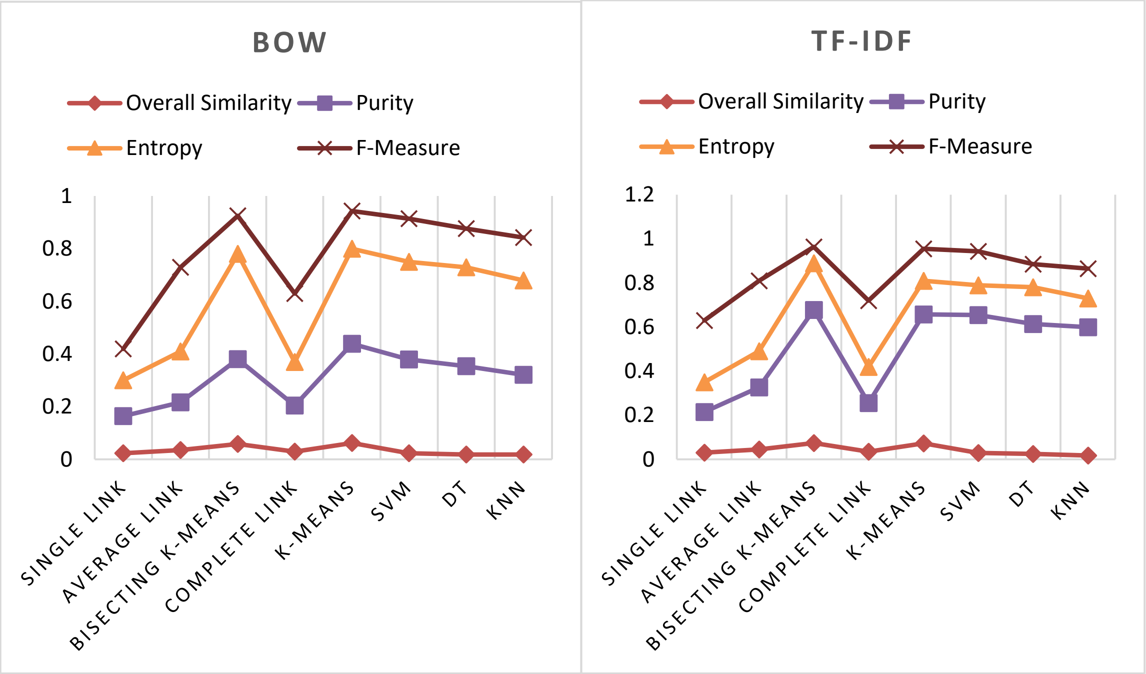


Figure 5.3.3: Quality of the clusters (k=3) obtained by the supervised and the unsupervised techniques

CHAPTER 5

Conclusion

In this study, we have presented significant findings to evaluate the classification of movie genres using textual features. As far as we know, this is the most extensive study done in this methodology. The proposed classification system is based on the supervised machine learning approaches are KNN, DT and SVM. At the same time, for the unsupervised machine learning approach, K-means clustering, Bisecting K-Means Clustering, and Agglomerative Hierarchical Clustering; average link, single link and the complete link are used. The algorithms have been combined with TF-IDF and BOW feature representation method and the dimensionality has been reduced by the Chi-Square feature selection method. After pre-processing phase, the prevailing supervised and unsupervised algorithms are implemented on our movie subtitle dataset. For unsupervised clustering, K-Means and Bisecting K-Means are more compatible than the Agglomerative Hierarchical Clustering in terms of quality of clusters using both feature representation methods. Agglomerative Hierarchical Clustering Algorithm usually produces inhomogeneous, unbalanced clusters. In contrast, for supervised document classification, SVM performs best when using the TF-IDF feature representation method. We also explored the performance with n-gram in terms of bigram trigram for unsupervised algorithms and we concluded that the use of N-Gram did not increase our model performance. In this study, the unsupervised and supervised techniques are compared in terms of cluster quality. We concluded that K-Means and Bisecting K-Means of unsupervised technique produced the cluster of better quality than KNN, SVM and DT in some cases, they are almost similar with SVM when using both feature representation method. Furthermore, unsupervised techniques produced clusters usually have greater overall similarity than the cluster produced by supervised techniques. These results validate the appearance of complementarity between supervised and unsupervised machine learning Technique to perform multilabel movie genre classification from movie subtitles.

REFERENCES

[1] H. Zhou, T. Hermans, A. V. Karandikar and J. M. Rehg, “Movie genre classification via scene categorization”, 18th ACM international conference on Multimedia, pages 747–750, 2010.

[2] G. Portolese and V. D. Feltrim, “On the use of synopsis-based features for film genre classification”, 15th National Meeting on Artificial and Computational Intelligence, pages 892–902, 2018.

[3] G. S. Simoes, J. Wehrmann, R. C. Barros and D. D. Ruiz, “Movie genre classification with Convolutional Neural Networks”, 2016 International Joint Conference on Neural Networks (IJCNN), 2016.

[4] C. Du, Z. Chin, F. Feng, L. Zhu, T. Gan and L. Nie, “Explicit Interaction Model towards Text Classification”, AAAI Conference on Artificial Intelligence, 33(01), 6359-6366, 2018.

[5] M. M. Mironczuk and J. Protasiewicz, “A recent overview of the state-of-the-art elements of text classification”, Expert Systems with Applications, 106, 36–54, 2018.

[6] B. Yu, “An evaluation of text classification methods for literary study”, Literary and Linguistic Computing, Volume 23, Issue 3, Pages 327–343, September 2008.

[7] R. Du, S. Naini and W. Susilo, “Web filtering using text classification”, The 11th IEEE International Conference on Networks, 2003.

[8] E. Amer and A. Nabil, “A Framework to Automate the generation of movies trailers using only subtitles”, 7th International Conference on Software and Information Engineering, Pages 126–130, 2018.

[9] A. M. Ertugrul and P. Karagoz, “Movie Genre Classification from Plot Summaries Using Bidirectional LSTM”, 2018 IEEE 12th International Conference on Semantic Computing (ICSC), 2018.

[10] R. B. Mangolin, R. M. Pereira, A. S. Britto, C. N. Silla, V. D. Feltrim, D. Bertolini and Y. M. G. Costa, “A multimodal approach for multi-label movie genre classification”, Multimedia Tools and Applications, 2020.

[11] S. C. Dharmadhikari, M. Ingle and P. Kulkarni, “Empirical Studies on Machine Learning Based Text Classification Algorithms”, Advanced Computing: An International Journal (ACIJ), Vol.2, No.6, November 2011.

[12] P. G. Shambharkar, P. Thakur, S. Imadoddin, S. Chauhan and M. N. Doja, “Genre Classification of Movie Trailers using 3D Convolutional Neural Networks”, 2020 4th International Conference on Intelligent Computing and Control Systems (ICICCS), 2020.

[13] H. Wang and H. Zhang, (2018), “Movie genre preference prediction using machine learning for customer-based information”, 2018 IEEE 8th Annual Computing and Communication Workshop and Conference (CCWC), 2018.

[14] J. A. Wi, S. Jang and Y. Kim, (2020), “Poster-Based Multiple Movie Genre Classification Using Inter-Channel Features”, IEEE Access, 8, 66615–66624, 2020.

[15] Y. F. Huang and S. H. Wang, “Movie Genre Classification Using SVM with Audio and Video Features”, Lecture Notes in Computer Science, 1–10, 2012.

[16] Badan Pengembangan dan Pembinaan Bahasa, Kemdikbud (Pusat Bahasa), Available at: https://kbbi.kemdikbud.go.id/entri/klasifikasi, Accessed: 22 December 2020.

[17] T. H. Susilo and S. Rochimah, "Classification of Topics and Analysis of Sentiments in Social Media", SNASTI, vol. 1, pp. 1–9, 2013.

[18] B. Hur and J. Weston, "A User's Guide to Support Vector Machines", Data Mining Techniques for the Life Sciences, 223–239, 2009.

[19] A. Geron, Hands-On Machine Learning with Scikit-Learn & TensorFlow, 7th Edition, pp: 67-179.

[20] E. H. Han, G. Karypis and V. Kumar, "Text categorization using weighted adjusted K-nearest neighbor classification", Lecture Notes in Computer Science, 53–65, May 1999.

[21] C. F. Eick, N. Zeidat and Z. Zhao, "Supervised clustering - algorithms and benefits", 16th IEEE International Conference on Tools with Artificial Intelligence, 2004.

[22] P. Berkhin, "A Survey of Clustering Data Mining Techniques. Grouping Multidimensional Data", 25–71, 2002.

[23] A. Mathew and M. Goswami, "A Comparative Study on Document Clustering Techniques, International Journal of Engineering Research & Technology", Volume 03, Issue 03, March 2014.

[24] YIFY Subtitles for English Movies, Available at: https://yts-subs.com/, Accessed: 10 December 2020.

[25] Sentiment Symposium Tutorial, Available: http://sentiment.christopherpotts.net/codedata/happyfuntokenizing.py, Accessed: 03 January 2020.

[26] Bitbucket, Available at: https://bitbucket.org/jaganadhg/twittertokenize/src/, Accessed: 12 March 2020.

[27] Natural language toolkit, Available at: http://www. nltk.org, Accessed: 23 June 2020.

[28] B. Yu, "An evaluation of text classification methods for literary study", Literary and Linguistic Computing, 23(3), 327–343, 2008.

[29] X. Hu and J. S. Downie, "Improving mood classification in music digital libraries by combining lyrics and audio", 10th Annual Joint Conference on Digital Libraries, pp. 159–168, 2010.

[30] M. F. Porter, "An algorithm for suffix stripping", Electronic Library and Information Systems, 14, 130–137, 1980.

[31] F. Sebastiani, "Machine learning in automated text categorization", ACM Computing Surveys, 34(1), 1–47, 2002.

[32] C. Moral, A. D. Antonio, R. Imbert and J. Ramirez, "A survey of stemming algorithms in information retrieval", Politecnica de Madrid, vol. 19 no. 1, March 2014.

[33] W. B. Cavnar and J. M. Trenkle, “N-gram-based text categorization”, 3rd Annual Symposium on Document Analysis and Information Retrieval, pp. 161–175, 1994.

[34] M. Zolotukhin and T. Hamalainen, “Detection of Zero-day Malware Based on the Analysis of Opcode Sequences”, 11th Consumer Communications and Networking Conference (CCNC), 2014.

[35] W. B. Carnar and J. M. Trenkle, “N-Gram-Based Text Categorization”, 3rd Annual Symposium on Document Analysis and Information Retrieval, pp: 161-175, 1994.

[36] A. K. Uysal, S. Gunal, S. Ergin and E. S. Gunal, “Detection of sms spam messages on mobile phones”, IEEE, pp. 8–11, 2012.

[37] C. D. Manning, P. Raghavan and H. Schutze, Introduction to information retrieval, Cambridge University Press, 2008, pp. 482.

[38] A. K. Uysal and S. Gunal, “A novel probabilistic feature selection method for text classification”, Knowledge-Based System, vol. 36, pp. 226–235, 2012.

[39] Y. Zhai, W. Song, X. Liu, L. Liu & X. Zhao, “A Chi-Square Statistics Based Feature Selection Method in Text Classification”, 2018 IEEE 9th International Conference on Software Engineering and Service Science (ICSESS), 2018.

[40] F. Thabtah, M. A. H. Eljinini, M. Zamzeer and M. Hadi, “Naive Bayesian Based on Chi Square to Categorize Arabic Data”, Communications, vol. 10, pp. 158–163, 2009.

[41] A. Ayedh, G. Tan, K. Alwesabi and H. Rajeh, “The Effect of Preprocessing on Arabic Document Categorization”, Algorithms, vol. 9, no. 2, 2016.

[42] I. Yoo and X. Hu, “A comprehensive comparison study of document clustering for a biomedical digital library MEDLINE”, 6th ACM/IEEE-CS Joint Conference on Digital Libraries, 2006.

[43] Y. Yang and X. Liu, “A Re-examination of Text Categorization Methods”, 22nd ACM International Conference on Research and Development in Information Retrieval, Pages 42–49, 1999.

Multilevel Movie Genre Classification from Movie Subtitle Using Supervised and Unsupervised Machine Learning Approach

ORIGINALITY REPORT

12%	%	%	12%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	Submitted to Universitas Atma Jaya Yogyakarta Student Paper	1%
2	Submitted to University of Information Technology, Yangon Student Paper	1%
3	Submitted to Universiti Kebangsaan Malaysia Student Paper	1%
4	Submitted to University of Melbourne Student Paper	1%
5	Submitted to University of Technology Student Paper	<1%
6	Submitted to Bridgepoint Education Student Paper	<1%
7	Submitted to Universiti Teknologi Malaysia Student Paper	<1%
8	Submitted to CSU, San Jose State University Student Paper	<1%