An Approach on Improving the Recommender System: Predicting Movie Genres Based on Plot Summaries

Gun Il Kim
Graduate School of Information
Yonsei University
Seoul, South Korea
kim_gunil_94@yonsei.ac.kr

Jae Heon Kim
Graduate School of Information
Yonsei University
Seoul, South Korea
jhk774@yonsei.ac.kr

Minkyoung Kim
Graduate School of Information
Yonsei University
Seoul, South Korea
minky@yonsei.ac.kr

Taekyoung Kwon
Graduate School of Information
Yonsei University
Seoul, South Korea
taekyoung@yonsei.ac.kr

Beakcheol Jang*
Graduate School of Information
Yonsei University
Seoul, South Korea
bjang@yonsei.ac.kr

Abstract— **This paper explores various machine learning methods such as logistic regression, support vector machine, and gradient-based variants to predict the multi-label and multi-class movie genres based on the plot summaries. To vectorize the plot summaries, two text representation methods are implemented including the Term-Frequency and Inverse-Document Frequency (TF-IDF) and Bag-of-Words (BoW) algorithms. The result on the comparison between the text representation models showed that the logistic regression model outperformed other machine learning models including stochastic gradient descent, support vector machine, and gradient-boosting variants, with the score of 0.504 in discounted cumulative gain score, and 0.628 in F1-score using TF-IDF approach.**

***Keywords*— multi-class, multi-label, term-frequency, inverse-document frequency, bag-of-words, machine-learning**

I. INTRODUCTION

Movies can be expressed as bringing delightful enjoyment in our lives as it plays an important role not only in the film industry but also, to the audience as a place to take part in socializing and for leisure. Some movies are straightforward and obvious; while others take a much deeper thought to consider such as in depicting the characters' style, checking the background history, and comprehending the expressive language that are presented throughout the story. These attributes can be perceived as metadata which brings descriptive facts such as synopsis, rating scores, actors and directors. Analyzing such complex data can be intriguing because we can make insightful predictions or classifications for better recommendation system to the audience and film industries. Therefore, this paper discusses methods in which we can improve the recommender system for movie recommendations through unique genres. This problem is based on the multi-label classification problem of movie genres from plot summaries. We evaluate the test dataset using several machine learning models such as logistic regression, stochastic gradient descent (SGD), support vector machine (SVM), and gradient-boosting variant models with two text vectorization methods including term-frequency and inverse-document frequency (TF-IDF) and bag-of-words (BoW) algorithm while predicting the multi-label genres. All models are evaluated using two metrics of nDCG (normalized discounted cumulative gain) and F1-score.

The rest of the paper is organized as follows. Section 2 discusses the related work based on our research. Section 3 introduces and explains the dataset and section 4 discusses a brief introduction of each machine learning models used in our experiment. Section 5 discusses the experiment results. Section 6 summarizes the conclusion and provide some future work.

II. RELATED WORK

According to several previous research, there had been some literature in solving similar problem to my project. Reference [5] discusses about predicting the preference of movie genres from customer-based information based on machine learning. The main idea is to develop a better movie recommendation system for small- and medium-sized enterprises (SMEs) through classification models from customer's demographic, behavioral and social information to predict their movie genre preference. Majority of users are usually consistent with their preferences as they prefer one genre over the other.

Reference [4], [7] and [8] experiment in predicting movie genres from movie posters based on deep learning approach. As movie posters provide an effective method of presenting meaningful information to the audience, it can determine their

* Corresponding author. E-mail: bjang@yonsei.ac.kr

interests and preferences. In fact, movie posters have strong visual attributes that can potentially attract many users to watch that specific movie over the other. Texture, color, actors, and their corresponding names are important features. By using Convolutional Neural Network (CNN) model, it captures multiple objects (i.e., person, car, bicycle, book, etc.) in every movie poster to pick out only the rich features in order to predict the movie genres, and by producing such model, it can potentially build a better recommender system for the film industry.

Reference [6] investigates movie trailers in predicting the movie genre based on the LSTM (Long-Short Term Memory) method. Like posters, video trailers can inform many users because it can present much more information realistically and lively. Using pre-trained model (i.e., VGG Net), the paper concludes with appealing results of 80.1% with spatial features, and 85% with LSTM from YouTube-Trailer dataset.

Along with previous methods, this project applies several machine learning algorithms on plot summaries in predicting the movie genre. As plot summaries are in text format, they must be converted into sequences of vectors and such conversion method is called text representation method. To vectorize the documents, we use two algorithms including TF-IDF (Term Frequency and Inverse-Document Frequency) and BoW (Bag of Words) methods because they provide efficient and effective results according to previous research papers [1][9]. Since movies can be labelled with several genres, this project is related to the multi-label classification problem. Logistic Regression (LR), Stochastic Gradient Descent (SGD), and Support Vector Machine (SVM), Light Gradient Boosting Model (LightGBM), and Extreme Gradient Boosting Model (XGB) are used as they are some of well-known models to be effective in text classification tasks.

III. Dataset

To prepare the dataset for our experiment, we extracted the movie metadata using Python's API to scrape directly from IMDB site over 18,000 movies from May 1st, 2000, to May 1st, 2020 with at least 100 votes from users for each movie. The original dataset held 18,538 movies, but after data preprocessing, the dataset was finalized with 17,973 movies. The genre types and each corresponding percentages are (in descending order): drama (23.9%), comedy (13.7%), action (8.79%), thriller (8.19%), romance (6.55%), crime (6.5%), horror (5.37%), adventure (4.97%), mystery (3.44%), biography (3.12%), history (2.67%), family (2.57%), fantasy (2.49%), sci-fi (1.99%), animation (1.97%), war (1.24%), music (1.18%), sport (0.73%), musical (0.43%), and western (0.19%). As the initial genre distribution showed a great imbalance among the top-20 genres, we grouped minor set of genres into top-5 genres, producing a balanced genre set. Top-5 genres are action (action/crime), comedy (comedy/romance), drama, horror (horror/mystery/thriller/fantasy/sci-fi) and miscellaneous (other remaining genres) types, each having around 6500 genres.

IV. Methods

A. *Term-Frequency and Inverse-Document Frequency*

Term-frequency and inverse-document frequency (TF-IDF) is simple but effective in vectorizing words into sequences of numeric vectors and it is still utilized in many domains as its baseline for its simplicity compared to other deep-learning state-of-the-art models in which it brings highly computational memory usage. As plot summaries sum up the full story for a much simpler description, it only grasps a few sentences or a paragraph length.

Given a set of documents, it counts the frequency of each word occurring for all sentences in a document (i.e., Term-Frequency) and also examines the occurrence rate for every word appearing in other documents (i.e., Inverse Document Frequency). For the first term, it counts the number of words in a sentence, and the frequency then becomes the sequence of vectors to represent the selected sentence. For every document, it produces n-words in each sentence length, producing a matrix factorization format. The more frequent a selected word becomes, the greater the count of that specific word, whereas if the chosen word is not visible in other documents, it is counted as zero. For the second term, it measures the uniqueness of a given word, in which the more common (i.e., increase in the number of counts) it appears in other documents, it becomes 0, otherwise 1 since it is comparatively distinctive to find in other documents. Mathematically, the equation is:

$$\text{tf-idf}(t, d, D) = \text{tf}(t,d) \cdot \text{idf}(t,D) \tag{1}$$

$$\text{tf}(t, d) = \log(1 + \text{freq}(t,d)) \tag{2}$$

$$\text{idf}(t, D) = \log\left(\frac{N}{\text{count}(d \in D : t \in d)}\right) \tag{3}$$

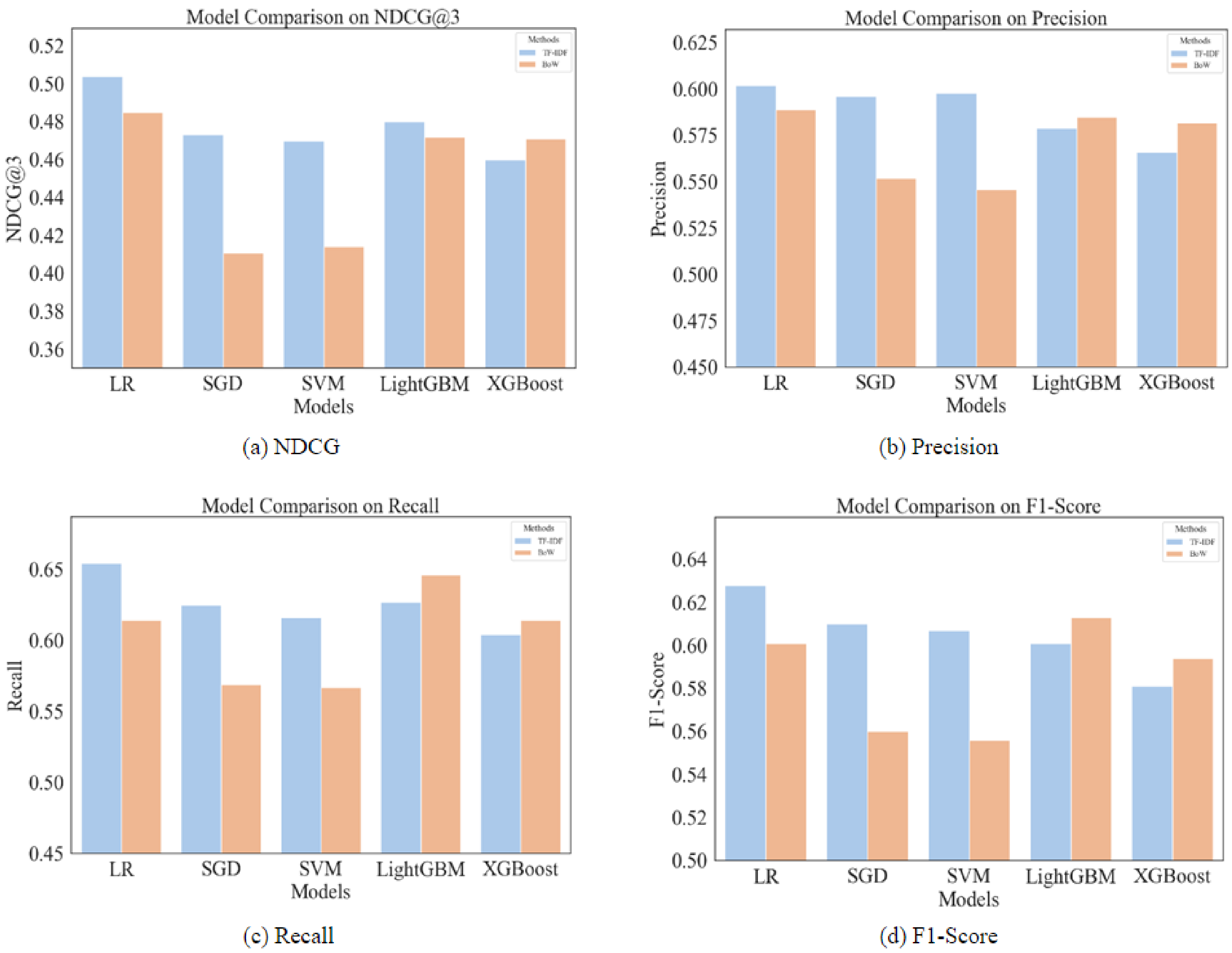
where freq(t,d) is the number of times term t appears in a document over total number of terms in the document; N is the total number of documents; and $\text{count}(d \in D : t \in d)$ is the number of documents with term t in it.

B. *Bag of Words*

Bag-of-words (BoW) is another model in extracting keywords and vectorizing documents into sequences of numbers. Compared to TF-IDF method, the sequence of steps is much different as it builds a library of vocabularies within the given document, seeks to find the maximum sentence length from the entire documents to produce a fixed-length document representation, and apply binary vector with 1 for having the word from the built-dictionary in the sentence, otherwise 0.

C. *Evaluation Metrics*

In information retrieval and recommender systems, there are several metrics that measure the performance of ranking quality in terms of the level of relevance. Recommending relevant information for each user plays a crucial role in information retrieval and recommender systems as irrelevant information diminishes user's satisfaction and preference. Cumulative Gain (CG) is one of the metrics that measure the relevance level that puts a greater emphasis on the most relevant document in the ranked list, assuming that it is more valuable than marginally relevant documents, and depending on the ranked position, the greater it becomes the less valuable for the user to eventually



(a) NDCG

(b) Precision

(c) Recall

(d) F1-Score

Figure 1: Graphs on model comparison between TF-IDF and BoW methods. (1a): Model comparison on nDCG@3 metric. (1b): Model comparison on Precision metric. (1c): Model comparison on Recall. (1d): Model comparison on F1-score.

look over the n-th positioned document. The cumulative gain is the sum of up to the ranked position i from 1 when i ranges between 1 to n. If we denote the position i in the gain vector G by G[i], the cumulative gain vector of i-th position is denoted as the vector CG[i] where,

$$\mathrm{CG[i]} = \left\{ \begin{array}{ll} G[1], & if\ i = 1 \\ CG[i-1] + G[i], & otherwise \end{array} \right\} \qquad (4)$$

To prevent such a dramatic increase from shared document score for every rank position added in CG, there must be an additional discounting formula for users to persist in examining further documents with a much-reduced score. In this case, the ideal method is to divide the document score by the log of its rank. This is called discounted cumulative gain (DCG), and it is an intuitively simple calculation. For example, $2_{log}2 = 1$ and $2_{log}512 = 9$, in which at position 512 is one ninth of its face value. Therefore, using the base of the logarithm can greatly provide much smaller the discounted score for every increase in the rank position. Mathematically, if b denotes the base from logarithm, the cumulative gain vector with discounted score is defined as:

$$\mathrm{DCG[i]} = \left\{ \begin{array}{ll} CG[i], & if\ \ i < b \\ DCG[i-1] + \frac{G[i]}{b_{logi}}, & if\ \ i \geq b \end{array} \right\} \qquad (5)$$

Comparing CG score with DCG score, there is a huge gap for every increase in the rank position (i.e., first 10 positions). As the order of relevance is also significant, we must construct an ideal possible relevance output to be compared with the actual relevance score. We define the ideal possible relevance score as IDCG, assuming o, l, and m relevant documents to be the relevant levels 1, 2, and 3 respectively, we calculate the following as the best ideal gain vector:

$$BV[i] = \left\{ \begin{array}{c} 3,\ \ if\ i\ \leq m, \\ 2,\ \ if\ m\ < i\ \leq m + l \\ 1,\ \ if\ m + l\ < i \leq m + l + o, \\ 0,\ \ otherwise. \end{array} \right\} \qquad (6)$$

Finally, by dividing the actual DCG score over IDCG score, we produce normalized DCG (nDCG) score in which it

Table 1: Summary of different methods of comparative machine learning model performance on discounted cumulative gain (DCG) and F1-scores. From top, LR: Logistic Regression, SGD: Stochastic Gradient Descent, LightGBM: Light Gradient Boosting Model, XGB: Extreme Gradient Boosting model.

Methods	Models	NDCG@3	Precision	Recall	F1-Score
TF-IDF (Unigram)	**LR**	**0.504**	**0.602**	**0.654**	**0.628**
	SGD	0.473	0.596	0.625	0.610
	SVM	0.47	0.598	0.616	0.607
	LightGBM	0.48	0.579	0.627	0.601
	XGBoost	0.46	0.566	0.604	0.581
BoW (Bag of Words)	**LR**	**0.485**	**0.589**	0.614	0.601
	SGD	0.411	0.552	0.569	0.560
	SVM	0.414	0.546	0.567	0.556
	LightGBM	0.472	0.585	**0.646**	**0.613**
	XGBoost	0.471	0.582	0.614	0.594

provides the percentage rate of the relevance level up to rank position 'K'.

$$\text{nDCG@K} = \text{DCG@K} / \text{IDCG@K} \qquad (7)$$

As the nDCG score increases to 1, the overall recommended documents are highly valuable for the user, otherwise near 0 represents ineffective relevance mark and therefore, the model performs poorly.

In addition, to further compare performance with different methods, we also calculated the F-score including precision and recall for both training and test data. F-score metric is defined as:

$$\text{Precision} = \frac{TP}{TP+FP} \qquad (8)$$

$$\text{Recall} = \frac{TP}{TP+FN} \qquad (9)$$

$$\text{F} - \text{score} = \frac{2 \text{ x } Precision \text{ x } Recall}{Precision+Recall} \qquad (10)$$

where, TP, FP, TN, and FN are True Positive, False Positive, True Negative and False Negative accordingly.

V. Results

During our experiment, we have tested two methods of text representation including term-frequency and inverse-document frequency (TF-IDF) and bag-of-words (BoW) and compared between machine learning models including logistic regression, stochastic gradient descent, support vector machine, and gradient-boosting variant models such as LightGBM and XGBoost. In TF-IDF approach, our model comparison showed that the logistic regression model outperformed other machine learning models with the scores of 0.504, 0.602, 0.654, and 0.628 in nDCG, precision, recall and F1-score respectively. Although the logistic regression showed the best performance in F1-metrics, it outperformed with a marginal difference, compared to other machine learning models such as SGD and SVM; while the nDCG score showed some great difference of 0.03, making the model pass solely 0.5 mark. Similarly, using the BoW method, the logistic regression model showed the best performance with 0.485 in nDCG score, despite of the LightGBM model to outperform in F1-score of 0.613. Fig. 1 provides some graphs of each metric including nDCG, precision, recall and F1-scores. Also, table 1 summarizes the experiment results of model comparison with different text vectorization techniques. Overall, comparing all different models, TF-IDF+logistic regression is the best choice as it presents the highest performance in all metrics.

VI. Conclusion and Future Work

This paper explores several machine learning methods being applied on movie summaries to predict the multi-label movie genres. This task is very challenging due to the small dataset as well as lack of descriptive plot summaries. One major flaw among all models is that it cannot understand the main context from the summary as some movies cannot convey the whole story to generate multiple genres from that specific movie. In respect to the test sets, it shows fair performance, particularly in logistic regression for TF-IDF and BoW methods. Although the model has shown some fair performance, we believe a future direction is to apply word embedding models such as Word2Vec model that can learn better sources of features. In addition, using pretrained embedding models can be a better option than traditional machine learning algorithms if the plot summary is descriptive and narrative.

Acknowledgment

This work was supported by the National Research Foundation of Korea (NRF) grants funded by the Korea government (MSIT) (No. RS-2023-00273751).

References

[1] A. Alahmadi, A. Joorabchi and A. E. Mahdi, "A New Text Representation Scheme Combining Bag-of-Words and Bag-of-Concepts Approaches for Automatic Text Classification", 2013 IEEE GCC Conference paper, University of Limerick, 2013.

[2] E. A. Makita and A. Lenskiy, "A Multinomial Probabilistic Model for Movie Genre Predictions", arXiv preprint arXiv:1603.07849, 2016.

[3] F. Colas and P. Brazdil, "Comparison of SVM and Some Older Classification Algorithms in Text Classification Tasks", in IFIP International Federation for Information Processing, Volume 217, Artificial Intelligence in Theory and Practice, ed. M. Bramer, (Boston: Springer), pp. 169-178, 2006.

[4] G. Barney and K. Kaya, "Predicting Genre from Movie Posters", Stanford University, 2019.

[5] H. Wang, H. Zhang, "Movie Genre Preference Prediction Using Machine Learning for Customer-Based Information", International Journal of Computer and Information Engineering, vol. 11, No.12, 2017.

[6] K. S. Sivaraman and G. Somappa, “MovieScope: Movie trailer classification using Deep Neural Networks”, Department of Computer Science, University of Virginia, 2017.

[7] N. Dave, “Predicting movie genres from movie posters”, Hampshire College, 2019.

[8] W. Chu and H. Guo, “Movie Genre Classification based on Poster Images with Deep Neural Networks”. In Proceedings of MUSA2’17, Mountain View, CA, USA, October 27, 2017.

[9] Z. Yun-tao, G. Ling and W. Yong-cheng, “An improved TF-IDF approach for text classification”, Department of Electronic & Information Technology, Shanghai Jiaotong University, 2004.

[10] Z. Wang, X. Sun, D. Zhang, and X. Li, “An Optimal SVM-Based Text Classification Algorithm”, 5th International Conference on Machine Learning and Cybernetics, 2006.