Predicting Quality and Popularity of a Movie From Plot Summary and Character Description Using Contextualized Word Embeddings

Jung-Hoon Lee
College of Computing
Sungkyunkwan University
Suwon-si, South Korea
vhrehfdl@gmail.com

You-Jin Kim
Algorima
Seoul-si, South Korea
k01077679687@gmail.com

Yun-Gyung Cheong
Department of AI,
Sungkyunkwan University
Suwon-si, South Korea
aimecca@skku.edu

***Abstract*—Narrative is an essential factor that makes games more enjoyable. However, predicting the story quality has been challenging for decades. In this paper, we propose to use contextual word embedding models such as BERT and ELMo, for predicting a story's success in terms of quality and popularity by using the story text only. Since deep learning models generally require extensive data, we conducted experiments to test the efficacy of our proposed model by leveraging the movie plot summaries. We present the results of the evaluations and conclude with discussions.**

***Index Terms*—contextual word embedding, deep learning, movie prediction, natural language processing, text classification**

I. Introduction

Narrative is one of the crucial factors that make the game more attractive[1]. While automated story generation has been researched for decades, evaluating the quality of a story is still challenging in the interactive environment.

On the other hand, in the movie industry, many works have been conducted to predict the story success. They have explored various factors such as social media opinions, investments, movie ratings, actors, release season, director, and movie genre. These features have been used to build machine learning and deep learning models to predict the success of a movie, as determined by the box office [1]–[8].

However, these metadata features are mostly available when the movie is produced and released. Therefore, it is difficult to use research results, using the above mentioned models, to estimate the quality and popularity of the story before production begins. To address this problem, a few studies have attempted to predict the box office before the movie is produced. [8] extracts features such as length of the scenario and the number of specific words appearing in the script. These features are employed to build machine learning models for movie success prediction. In our previous work [9], CMU plot summary corpus [10] is used to build deep learning models. The study hypothesized that the use of sentiment scores extracted from the summary can improve the prediction of a movie's success.

The aim of this paper is to create deep learning based classification models to predict movie success using only the movie scripts and synopses. To this end, we collected plot summaries and synopses from the IMDB site, which contains movies that have already been released[2]. The plot summary describes a movie within 250 words avoiding any spoilers. Since IMDB offers multiple plot summaries, we simply use the top summary among them. The synopsis is a more detailed description of a movie, possibly containing some spoilers.

Then, we classify a released movie's success, relying on the scores provided at the Rotten Tomatoes site[3]. The website offers two types of scores: audience score and Tomatometer. The audience scores are rated by the general movie viewers; hence, we define movie popularity based on the audience score. In a similar fashion, we determine a movie's quality relying on the Tomatometer score, as it is computed by the ratings given by hundreds of movie and television critics.

We hypothesize that the character description would enhance the performance of the model for predicting movie quality and popularity. We extract the sentences mentioning the main characters of the movie from the synopsis. Using the plot summary and parts of synopsis as input, our models classify the labels (successful or not successful) in terms of popular and qualitative movies.

The contributions of our research are as follows:

- We built a movie dataset consisting of the latest movies from 2000 to 2018 along with the plot summary and movie synopsis.
- We propose and implement deep learning models which use the plot summary and the character description of a movie to predict its popularity and quality.
- We evaluate the efficacy of the latest embedding models (BERT and ELMo) for movie success prediction.

[1]Game Skinny, February 1, 2018, https://www.gameskinny.com/b7099/the-importance-of-story-based-games

[2]https://www.imdb.com/

[3]https://www.rottentomatoes.com/

978-1-7281-4533-4/20/$31.00 ©2020 IEEE

TABLE I
TRAINING AND TEST DATA SET PROPORTION.
CLASS 1 DENOTES MOVIES WITH SCORES GREATER THAN OR EQUAL TO 75.
CLASS 0 DENOTES MOVIES WITH SCORES LESS THAN 60.

	Train		**Test**		
Genre/Quality	**Quality(1)**	**No Quality(0)**	**Quality(1)**	**No Quality(0)**	**Total**
Drama	959 (51%)	972 (49%)	114 (53%)	101 (47%)	2,146 (50%)
Comedy	456 (50%)	451 (50%)	48 (48%)	53 (53%)	1,008 (24%)
Action	262 (51%)	254 (49%)	25 (43%)	33 (56%)	574 (13%)
Thriller	241 (50%)	237 (50%)	25 (54%)	29 (46%)	532 (13%)
Subtotal	1,918	1,914	212	216	4,260
Genre/Popularity	**Popularity(1)**	**No Popularity(0)**	**Popularity(1)**	**No Popularity(0)**	
Drama	875 (50%)	881 (50%)	101 (52%)	95 (48%)	1,952 (51%)
Comedy	364 (50%)	359 (50%)	38 (47%)	43 (53%)	804 (21%)
Action	285 (51%)	278 (49%)	28 (43%)	35 (56%)	626 (17%)
Thriller	199 (50%)	198 (50%)	22 (49%)	23 (51%)	442 (11%)
Subtotal	1,723	1,716	189	196	3,824

- We applied the learned models to game stories for story popualrity prediction.

The section 2 describes our data. In Section 3, we suggest our proposed method. In Section 4, we discuss the evaluation and results. Finally, we conclude with future work.

II. DATA

A. Data Collection and Features

To build the dataset, we collected movie titles, release dates, plot summaries, and synopsis from the IMDB site. The Tomatometer scores and audience scores were collected from the Rotten Tomatoes website. In order to extract the character description part, we obtain up to three of the main characters' names. The plot summary is collected using the storyline section of IMDB, which consists of up to 250 words. The average word count of a synopsis was 1,043.

We crawled the data from Rotten Tomatoes and IMDB using the Selenium [4] and Beautiful Soup [11] python packages. The following shows an example, extracted for the movie 'The Avengers: Infinity War' directed by Anthony and Joe Russo (released in 2018).

- Movie title : Avengers Infinity War
- Release year : 2018
- Character : Tony Stark, Thor, Bruce Banner
- Genre : Action, Adventure, Sci-Fi
- Plot summary : As the Avengers and their allies have continued to protect the world from threats too large for any one hero to handle, a new danger has emerged from the cosmic shadows: Thanos. A despot of intergalactic infamy, his goal is to collect all six Infinity Stones, artifacts of unimaginable power, and use them to inflict his twisted will on all of reality. Everything the Avengers have fought for has led up to this moment, the fate of Earth and existence has never been more uncertain.
- Synopsis : Thanos and his Children - Proxima Midnight, Ebony Maw, Corvus Glaive and Cull Obsidian - have attacked the Asgardian ship in search of the Space Stone, which is housed in the Tesseract that Loki had stolen before Asgard's destruction.
  ...
  Former S.H.I.E.L.D. Director Nick Fury and Deputy Director Maria Hill witness the scene on the street before they dissolve themselves. Before he vanishes, Fury manages to send a final distress signal to Captain Marvel.
- Tomatometer : 85
- Audience score : 91

B. Data Labelling

For classification, we label the movies as 'successful' and 'not successful' in terms of popularity and quality, as mentioned in Section 1. Similar to the guideline of Rotten Tomatoes which classifies a movie as certified fresh (75 or higher) or rotten (less than 60), we determine popularity based on the audience score; a score greater than or equal to 75 means that the movie is 'popular', and a score less than or equal to 60 indicates that a movie is 'unpopular'. For the quality of a movie, we use the Tomatometer score; if a score is greater than or equal to 75, the movie is classified as 'qualitative'; if a movie score is less than or equal to 60, we classify it as 'non-qualitative'.

C. Data Analysis

After labeling the collected movies as successful or not, in terms of popularity and quality, we obtained 3,824 movies for popularity prediction and 4,260 movies for quality prediction. The two numbers differ since some of the movies lack either the audience score or the Tomatometer.

The number of movies per genre is listed in Table 1. We divide them into a train set and a test set, with the ratio of 9:1. Class imbalance is observed in the genre distribution. Drama occupies the largest portion, followed by comedy. These two genres account for 74% of the total number of movies

[4]https://www.seleniumhq.org/

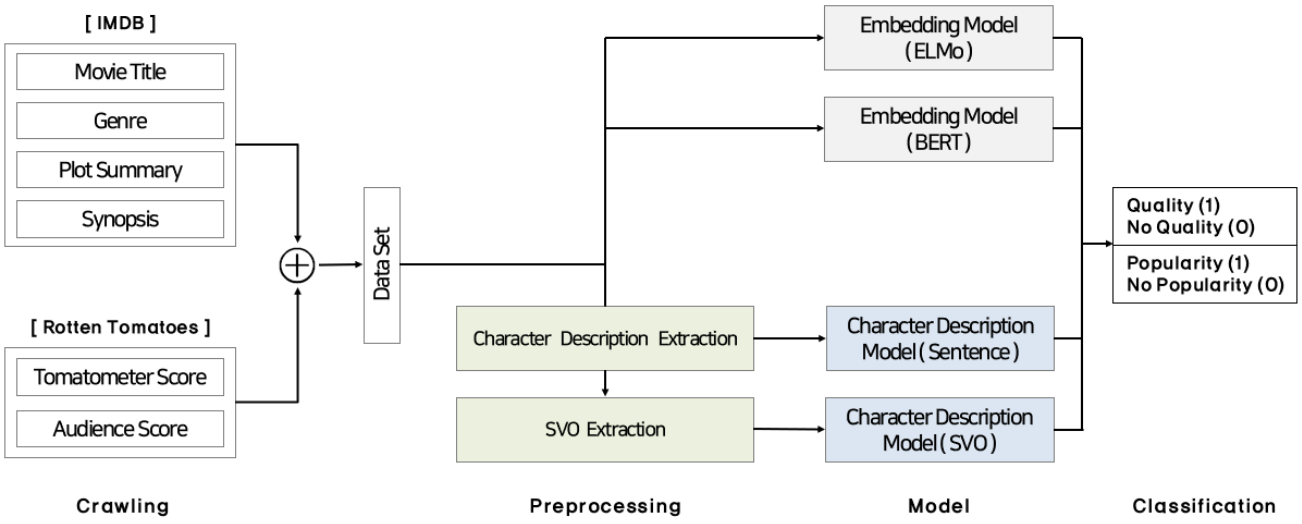

Fig. 1. The overall process of movie success prediction.

with quality labels and account for 72% of the movies with popularity labels. The ratios between 1 (quality, popularity) and 0 (no quality, no popularity) are balanced across all the genres.

The maximum number of words in a plot summary is 205 words. The average number of words in a summary is 93 for the data with quality labels, and 944 for the data with popularity labels. The maximum number of words in a synopsis is 11,396 words. The average number of words for the data in the quality prediction group is 1,036, and 1,043 for the data in the popularity prediction group.

III. Prediction Models

This section presents contextual embedding based approaches which exploit movie plot summaries and synopses, as shown in Figure 1. First, we collected the movie data from the IMDB and Rotten Tomatoes sites. Then, we construct the data from the crawled data. Third, we pre-processed the data to extract text which mentions the characters and their subject, verb, and object constituents in the extracted sentence. Finally, we train two different types of models for comparison: embedding models and character description models.

A. Embedding Models

Embedding Models primarily use contextual word embedding techniques for representing a plot summary (Figure 2). Word embedding is a method of transforming text data into numerical vectors for training deep learning models. Traditional word embedding models such as Word2vec [12], Fasttext [13] and Glove [14] produce a fixed vector for each word. Therefore, these models cannot generate different vectors for homophones words whose meanings vary depending on the context.

Recently, contextualized embedding methods have been devised that can generate different word vectors depending on the context. ELMo [15] and BERT [16] are effective contextualized embedding models, pre-trained with large amounts of

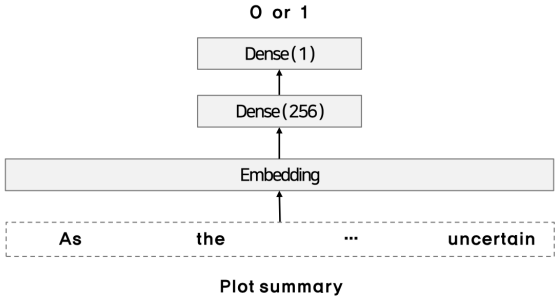

Fig. 2. Embedding Model architecture

training data. ELMo is an embedding method that uses two bidirectional LSTM networks to construct a vector. BERT is a method of embedding input data using only the encoder part of Transformer model [17].

In order to test the efficacy of embedding models, we build two prediction models which rely on ELMo and BERT. Embedding(ELMo) uses word representations generated by the ELMO embedding. A 1024 dimensional ELMO embedding vector is constructed for a plot summary containing a maximum of 250 words. Then, the vector is put into the 256 dimensional dense networks using RELU [18] as its activation function. Then, the vector is connected to the output layer with the sigmoid function to perform the binary classification.

The Embedding(BERT) model uses BERT as its embedding layer. A 768 dimensional BERT embedding vector is constructed for each summary, which is put into the 256 dimensional dense networks with GELU [19] as its activation function.

In this work, we utilized the TensorFlow Hub implementation [5] to represent the word embedding vectors. We fine-tuned

[5]https://tfhub.dev/google/elmo/2

the embedding models with our dataset for performance improvement [20].

B. Character Description Models

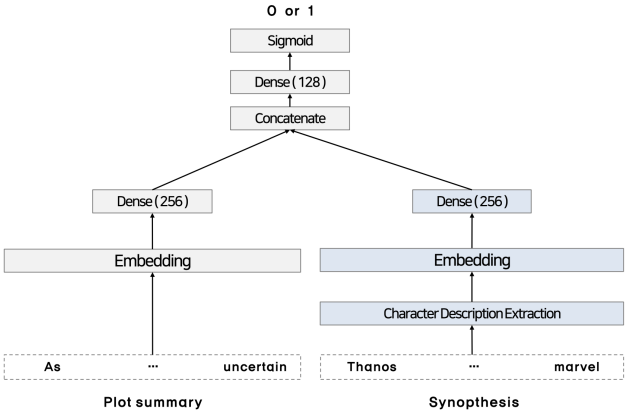

Fig. 3. Character Description (Sentence) Model Architecture

Character description is a set of sentences representing the personality of and information about a movie character [21]–[23]. We anticipate the information specific to a movies's main characters may enhance the movie success prediction.

We extracted the sentences containing the characters' names from the synopsis. This text is used as an input to the Character Description (Sentence) model. Then, we use the Spacy [6] and Textacy [7] libraries to extract a subject, a verb, and an object from each sentence in the character description. Table II shows an example extracted from the movie 'The Avengers: Infinity War'(released in 2018) directed by Anthony and Joe Russo.

TABLE II
CHARACTER DESCRIPTION EXAMPLES.
THE LEFT COLUMN SHOWS THE SENTENCES CONTAINING THE MAIN CHARACTERS, WHILE THE RIGHT COLUMN SHOWS THEIR CORRESPONDING SVO FORMS.

Character Description	SVO
After a futile counter attack from the Hulk, Loki offers the Tesseract to Thanos in exchange for Thor's life only to be killed himself when Thanos anticipates Loki's attempt to betray and kill him. Moments before Glaive kills him, Heimdall uses the power of the Bifrost to send Hulk to Earth.	Loki offers Tesseract. Thanos anticipates attempt. Glaive kills him.
Hulk crashlands at the Sanctum Sanctorum and is reverted back to Bruce Banner, who informs Stephen Strange and Wong about Thanos' impending arrival.	who informs Strange. who informs Wong.

As the table shows, the SVO extraction functionality is not perfect, especially when the sentence is complex. Pronoun resolution is not handled in this study. In some synopses, no sentence was extracted when the synopsis contains no character names. For instance, for popularity prediction in the thriller genre, no SVO was extracted for 41 out of the 442 movies.

1) Sentence and SVO Models: After extracting character descriptions and their corresponding SVOs, these are used along with the plot summary to build two models: Character Description Model (Sentence) and Character Description Model (SVO). Figure 3 illustrates Character Description (Sentence) Model. Using BERT Embedding, the plot summary is converted to a 768-dimensional vector and then is reduced to 256 dimensions via a dense layer. The character descriptions are also vectorized using BERT and are compressed to 256 dimensions. These vectors are concatenated to create a 512-dimensional vector. Next, we reduce the vector to a 128 dimensional representation to perform binary classification with the application of the sigmoid function.

Character Description Model (SVO) is similar to the Character Description Model (Sentence). The only difference lies in using the extracted SVOs of character description as input to the BERT Embedding layer.

IV. EVALUATION

This section reports our evaluations of the five different models including a benchmark model proposed in [9]. In order to compare our approach with previous work, we applied a sentiment model proposed in [9].

We replicated the benchmark model because the data used were different. The movies found in CMU plot summary corpus [10] were released in the 20th century. Therefore, many of these movies lack synopsis, the audience score, or the Tomatometer. To build the model, we first extracted the sentiment score from -1 (most negative) to 1 (most positive) for each sentence using NLTK's Vader sentiment analyzer. The sentiment score sequence vector is given to the bidirectional LSTM layers with 128 units. The outputs of these layers are added and flattened to create a 50,688 dimensional vector. We then concatenate a 50,688 dimensional vector created using the sentiment score and a 256 dimensional vector created using the plot summary. Then, the next 128 dense layer reduces the vector for the final binary classification. As in [9] we used the binary cross-entropy as the loss function and the Adam optimizer.

A. Quality Prediction

Table III shows the performance results for quality prediction, in terms of recall, precision, F1 scores, and accuracy. We obtained the highest accuracy of 0.70 for two genres: comedy and action. The highest F1 scores range from 0.54 (non-quality for drama) to 0.75 (non-quality for action). This reveals that it is relatively difficult to predict non-qualitative dramas.

When inspecting the classification models, Embedding(BERT) achieved the highest accuracy in action (0.70) and comedy (0.70). With the model, the F1 scores of predicting non-successful labels (0.75 for action, 0.72 for comedy) are higher than those of predicting successful labels (0.65 for action and 0.68 for comedy).

[6]https://spacy.io
[7]https://pypi.org/project/textacy

TABLE III
THE EVALUATION RESULTS FOR QUALITY PREDICTION IN PRECISION, RECALL, F1, AND ACCURACY.
THE BEST PERFORMANCES IN ACCURACY ARE IN BOLD.
LABEL 1 DENOTES SUCCESSFUL AND 0 DENOTES UNSUCCESSFUL.

Score	Genre	Model	Precision		Recall		F1		Accuracy
			1	**0**	**1**	**0**	**1**	**0**	
Quality	**Drama**	Embedding(ELMo)	0.58	0.56	0.72	0.41	0.64	0.47	0.55
		Embedding(BERT)	0.62	0.54	0.49	0.66	0.59	0.55	0.57
		Character Description(Sentence)	0.59	0.53	0.56	0.56	0.58	0.55	0.56
		Character Description(SVO)	**0.61**	**0.60**	**0.70**	**0.50**	**0.65**	**0.54**	**0.59**
		Sentiment Model [9]	0.55	0.51	0.68	0.39	0.61	0.44	0.52
	Comedy	Embedding(ELMo)	0.66	0.68	0.48	0.77	0.55	0.69	0.62
		Embedding(BERT)	**0.70**	**0.71**	**0.67**	**0.74**	**0.68**	**0.72**	**0.70**
		Character Description(Sentence)	0.59	0.73	0.79	0.51	0.68	0.60	0.64
		Character Description(SVO)	0.66	0.71	0.69	0.68	0.67	0.69	0.68
		Sentiment Model [9]	0.67	0.68	0.62	0.72	0.65	0.70	0.67
	Action	Embedding(ELMo)	0.57	0.70	0.64	0.64	0.60	0.67	0.63
		Embedding(BERT)	**0.67**	**0.74**	**0.64**	**0.76**	**0.65**	**0.75**	**0.70**
		Character Description(Sentence)	**0.67**	**0.74**	**0.64**	**0.76**	**0.65**	**0.75**	**0.70**
		Character Description(SVO)	0.73	0.64	0.32	0.91	0.44	0.75	0.59
		Sentiment Model [9]	0.58	0.74	0.72	0.61	0.64	0.67	0.65
	Thriller	Embedding(ELMo)	0.53	0.56	0.32	0.76	0.40	0.65	0.52
		Embedding(BERT)	0.67	0.56	0.16	0.93	0.26	0.70	0.48
		Character Description(Sentence)	**0.63**	**0.75**	**0.76**	**0.62**	**0.69**	**0.68**	**0.68**
		Character Description(SVO)	0.55	0.56	0.24	0.83	0.33	0.67	0.50
		Sentiment Model [9]	0.55	0.59	0.44	0.69	0.49	0.63	0.56

TABLE IV
THE EVALUATION RESULTS FOR POPULARITY PREDICTION IN PRECISION, RECALL, F1, AND ACCURACY.
THE BEST PERFORMANCES IN ACCURACY ARE IN BOLD.
LABEL 1 DENOTES SUCCESSFUL AND 0 DENOTES UNSUCCESSFUL.

Score	Genre	Model	Precision		Recall		F1		Accuracy
			1	**0**	**1**	**0**	**1**	**0**	
Popularity	**Drama**	Embedding(ELMo)	0.53	0.50	0.51	0.52	0.52	0.51	0.51
		Embedding(BERT)	0.58	0.58	0.67	0.47	0.62	0.52	0.57
		Character Description(Sentence)	0.56	0.59	0.78	0.34	0.65	0.43	0.54
		Character Description(SVO)	**0.65**	**0.58**	**0.50**	**0.72**	**0.57**	**0.64**	**0.60**
		Sentiment Model [9]	0.52	0.49	0.55	0.46	0.54	0.48	0.51
	Comedy	Embedding(ELMo)	0.55	0.61	0.58	0.58	0.56	0.60	0.58
		Embedding(BERT)	0.65	0.58	0.29	0.86	0.40	0.69	0.54
		Character Description(Sentence)	0.50	0.58	0.66	0.42	0.57	0.49	0.53
		Character Description(SVO)	**0.72**	**0.60**	**0.34**	**0.88**	**0.46**	**0.72**	**0.59**
		Sentiment Model [9]	0.51	0.58	0.61	0.49	0.55	0.53	0.54
	Action	Embedding(ELMo)	0.67	0.67	0.50	0.80	0.57	0.73	0.65
		Embedding(BERT)	**0.75**	**0.70**	**0.54**	**0.86**	**0.63**	**0.77**	**0.70**
		Character Description(Sentence)	0.90	0.64	0.32	0.97	0.47	0.77	0.62
		Character Description(SVO)	0.67	0.72	0.64	0.74	0.65	0.73	0.69
		Sentiment Model [9]	0.73	0.65	0.39	0.89	0.51	0.75	0.63
	Thriller	Embedding(ELMo)	0.64	0.70	0.73	0.61	0.68	0.65	0.66
		Embedding(BERT)	**0.69**	**0.79**	**0.82**	**0.65**	**0.75**	**0.71**	**0.73**
		Character Description(Sentence)	0.60	0.60	0.55	0.65	0.57	0.63	0.60
		Character Description(SVO)	**0.75**	**0.72**	**0.68**	**0.78**	**0.71**	**0.75**	**0.73**
		Sentiment Model [9]	0.70	0.68	0.64	0.74	0.67	0.71	0.69

Character Descriptions (Sentence) model also achieved the highest accuracy in two genres: action (0.70) and thriller (0.68). Character Description (SVO) model outperforms the other models in accuracy for the drama genre. However, its accuracy (0.59) and F1 scores (0.65 for successful and 0.54 for unsuccessful) are poor for practical usage.

B. Popularity Prediction

Table IV reports the results of popularity prediction. The highest accuracy of 0.73 was obtained for the thriller genre using Embedding (BERT) and Character Description (SVO) models. The highest F1 scores range from 0.46 (popular comedies) to 0.77 (unpopular action movies).

As observed in quality prediction, the performance scores of unsuccessful movies are higher than those of successful ones except for the thriller genre.

Overall, Character Description(SVO) and Embedding(BERT) outperform the other models. Embedding(BERT) achieved the highest accuracy in the action and thriller genres. It should be particularly noted that the model achieved the precision of 0.79 for non-successful movies in the thriller genre. This means that if this model filters out a movie script as 'not successful', 79% of the prediction is correct.

Character Description(SVO) outperform the other models in the drama and comedy genres. However, the F1 scores are low, especially for predicting successful comedies (0.46) and successful dramas (0.57). This suggests that a movie script can serve as a predictor for popularity in the action and thriller genres, but not for the drama and comedy. This model also achieved the highest accuracy in thriller (0.73); its F1 scores are 0.75 for predicting unpopular and 0.71 for predicting popular movies.

C. Discussions

Overall, the performance of predicting 'not successful' movies was higher than that of predicting 'successful' movies. When inspecting the results genre-wise, the thriller and action genres show higher performances than the drama genre.

We obtained the highest accuracy using Embedding (BERT) in four categories: quality prediction for comedy and action, and popularity prediction for action and thriller. This means that the BERT embedding model can represent the plot summary better than the ELMo embedding model for predicting a movie's success. It also indicates that sentential features can serve as an efficient predictor for the task.

The Character Description (SVO) models achieved the best performance in the four categories, both popular and quality predictions in drama, and popularity predictions in comedy and thriller. Character Description (Sentence) achieved the best performance for quality prediction in action and thriller. This confirms our hypothesis that the use of character description can improve prediction performance.

The Sentiment Model (LSTM) proposed in [9] did not achieve the best performance in any of the categories. Further investigation is needed to test whether the sentiment flow information would help better predict a movie success.

D. Game narrative applied

We applied the model trained using movie plot and synopsis to the game narrative. First, we collected the top 5 game stories on the Ranker website[8]. Ranker is a homepage that shows the results of users voting on the most entertaining and involving storylines. The game title ranked first is 'The Last of Us', which obtained 8,879 recommendations; the second to the fifth titles are the 'Red Dead Redemption', 'Star Wars: Knights of the Old Republic', 'BioShock Infinite' and 'Fallout: New Vegas games'.

Then we chose five game titles that were deprecated more than recommended. 'Neverwinter Nights' is a game that received 337 recommendations and 445 deprecations, ranked as the 121st place. The other mostly deprecated game titles are 'Sleeping Dogs', 'Castlevania: Symphony of the Night', 'Devil May Cry 3', 'Left 4 Dead' and 'Saints Row: The Third'.

We only predict the game story's popularity because the rankings at Ranker are based on the gamers' ratings. Moreover, the majority of the game titles belong to the action genre. Therefore, among the models learned for popularity classification, we chose three models built for the movie's action genre: Embedding (BERT), Sentiment Model, and Character Description (SVO), for their applications to the game story. The experimental results are shown in the table V.

[8]https://www.ranker.com/crowdranked-list/the-most-compelling-video-game-storylines

TABLE V
GAME STORY APPLY RESULT
THE TOP FIVE ARE POPULAR GAME STORIES,
AND THE BOTTOM FIVE ARE NOT POPULAR GAME STORIES.

Game Title	Embedding (BERT)	Sentiment Model	Character Description(SVO)
The Last of Us	No popularity	No popularity	Popularity
Red Dead Redemption	No popularity	No popularity	Popularity
Star Wars	Popularity	No popularity	Popularity
BioShock Infinite	No popularity	No popularity	Popularity
Fallout: New Vegas	Popularity	No popularity	Popularity
Neverwinter Nights	No Popularity	No popularity	Popularity
Sleeping Dogs	No Popularity	popularity	Popularity
Castlevania: Symphony of the Night	Popularity	No popularity	Popularity
Devil May Cry 3	No Popularity	No popularity	Popularity
Left 4 Dead	No Popularity	No popularity	Popularity

The results show that the classification results are very different depending on the model applied. The Embedding(BERT) model classifies only two of the top five game titles as *popular*, while it accurately classifies all of the five low ranked game titles as *not popular*. The Sentiment model shows the worst performance, classifying all game titles as *not popular*, except one of the low ranked games–"Sleeping Dogs". The model tends to classify game stories as *not popular*. We believe that the lack of emotional expression in the game story leads to this result. The Character Description(SVO) classifies all game stories as *popular*. The results suggest that the model is not suitable for game story classification. We suspect that the model is overfitted to movie stories and their character descriptions.

V. CONCLUSIONS

In this paper, we propose word embedding based approaches that use plot summary and character description for predicting movie success in terms of popularity and quality. To evaluate the performance of the proposed models, we prepared data sets: movie plot summaries gathered from the IMDB and their review scores from Rotten Tomatoes. We built four models using BERT and ELMo embeddings, and character description sentencess.

We obtained the highest accuracy of 0.73 for predicting popularity of a movie in the thriller genre. The highest accuracy for predicting movie quality was 0.70 for the action genre. The evaluation results show that the use of BERT embedding for summary representation is more effective than ELMo embedding. The results also indicate that the addition of character description can improve the performance.

The evaluation results are promising, considering that only textual summaries of the movie plot are used, without having any prior information such as director, cast, budget, which are critical for the success of a movie.

The results also suggest that predicting unsuccessful movies performs better than that of predicting successful movies. Therefore, the models presented in this paper can be useful for filtering out movie scripts that may not appeal to the audience. Furthermore, we applied the models to several game narratives and obtained preliminary results. For future work, we will extend the dataset to include game narratives. We plan to experiment with the recently developed embedding models, such as XLNet. We hope that the proposed method can facilitate the decision-making in funding movies and game productions.

VI. Acknowledgments

This research was partly supported by Basic Science Research Program through the National Research Foundation of Korea(NRF) funded by the Ministry of Education(2019R1A2C1006316) and Institute of Information communications Technology Planning Evaluation(IITP) grant funded by the Korea government(MSIT) (No.2019-0-00421, AI Graduate School Support Program).

References

[1] J. Du, H. Xu, and X. Huang, "Box office prediction based on microblog," *Expert Systems with Applications*, vol. 41, no. 4, pp. 1680–1689, 2014.

[2] S. R. Jaiswal and D. Sharma, "Predicting success of bollywood movies using machine learning techniques," in *Proceedings of the 10th Annual ACM India Compute Conference on ZZZ*. ACM, 2017, pp. 121–124.

[3] K. Lee, J. Park, I. Kim, and Y. Choi, "Predicting movie success with machine learning techniques: ways to improve accuracy," *Information Systems Frontiers*, vol. 20, no. 3, pp. 577–588, 2018.

[4] T. G. Rhee and F. Zulkernine, "Predicting movie box office profitability: a neural network approach," in *2016 15th IEEE International Conference on Machine Learning and Applications (ICMLA)*. IEEE, 2016, pp. 665–670.

[5] L. Zhang, J. Luo, and S. Yang, "Forecasting box office revenue of movies with bp neural network," *Expert Systems with Applications*, vol. 36, no. 3, pp. 6580–6587, 2009.

[6] R. Sharda and D. Delen, "Predicting box-office success of motion pictures with neural networks," *Expert Systems with Applications*, vol. 30, no. 2, pp. 243–254, 2006.

[7] Y. Zhou, L. Zhang, and Z. Yi, "Predicting movie box-office revenues using deep neural networks," *Neural Computing and Applications*, vol. 31, no. 6, pp. 1855–1865, 2019.

[8] J. Eliashberg, S. K. Hui, and Z. J. Zhang, "Assessing box office performance using movie scripts: A kernel-based approach," *IEEE Transactions on Knowledge and Data Engineering*, vol. 26, no. 11, pp. 2639–2648, 2014.

[9] Y. J. Kim, J. H. Lee, and Y.-G. Cheong, "Prediction of a movie's success from plot summaries using deep learning models," in *Proceedings of the Second Workshop on Storytelling*, 2019, pp. 127–135.

[10] D. Bamman, B. O'Connor, and N. A. Smith, "Learning latent personas of film characters," in *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Sofia, Bulgaria: Association for Computational Linguistics, Aug. 2013, pp. 352–361. [Online]. Available: https://www.aclweb.org/anthology/P13-1035

[11] L. Richardson, "Beautiful soup," *Crummy: The Site*, 2013.

[12] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," *arXiv preprint arXiv:1301.3781*, 2013.

[13] P. Bojanowski, E. Grave, A. Joulin, and T. Mikolov, "Enriching word vectors with subword information," *Transactions of the Association for Computational Linguistics*, vol. 5, pp. 135–146, 2017.

[14] J. Pennington, R. Socher, and C. Manning, "Glove: Global vectors for word representation," in *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, 2014, pp. 1532–1543.

[15] M. E. Peters, M. Neumann, M. Iyyer, M. Gardner, C. Clark, K. Lee, and L. Zettlemoyer, "Deep contextualized word representations," *arXiv preprint arXiv:1802.05365*, 2018.

[16] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," *arXiv preprint arXiv:1810.04805*, 2018.

[17] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in neural information processing systems*, 2017, pp. 5998–6008.

[18] V. Nair and G. E. Hinton, "Rectified linear units improve restricted boltzmann machines," in *Proceedings of the 27th international conference on machine learning (ICML-10)*, 2010, pp. 807–814.

[19] D. Hendrycks and K. Gimpel, "Gaussian error linear units (gelus)," *arXiv preprint arXiv:1606.08415*, 2016.

[20] C. S. Perone, R. Silveira, and T. S. Paula, "Evaluation of sentence embeddings in downstream and linguistic probing tasks," *arXiv preprint arXiv:1806.06259*, 2018.

[21] P.-E. Mazaré, S. Humeau, M. Raison, and A. Bordes, "Training millions of personalized dialogue agents," *arXiv preprint arXiv:1809.01984*, 2018.

[22] H. Kim, D. Katerenchuk, D. Billet, J. Huan, H. Park, and B. Li, "Understanding actors and evaluating personae with gaussian embeddings," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, 2019, pp. 6570–6577.

[23] J. Li, M. Galley, C. Brockett, G. P. Spithourakis, J. Gao, and B. Dolan, "A persona-based neural conversation model," *arXiv preprint arXiv:1603.06155*, 2016.