Copycat vs. Original: Multi-modal Pretraining and Variable Importance in Box-office Prediction

Qin Chao[1,3], Eunsoo Kim[2], and Boyang Li[1]
chao0009@ntu.edu.sg, eunsoo@uos.ac.kr, boyang.li@ntu.edu.sg
[1]*College of Computing and Data Science, Nanyang Technological University, Singapore*
[2]*Business School, University of Seoul, Republic of Korea*
[3]*Alibaba Group and the Alibaba-NTU Joint Research Institute, Singapore*

***Abstract*—The movie industry is associated with an elevated level of risk, which necessitates the use of automated tools to predict box-office revenue and facilitate human decision-making. In this study, we build a sophisticated multimodal neural network that predicts box offices by grounding crowdsourced descriptive keywords of each movie in the visual information of the movie posters, thereby enhancing the learned keyword representations, resulting in a substantial reduction of 14.5% in box-office prediction error. The advanced revenue prediction model enables the analysis of the commercial viability of "copycat movies," or movies with substantial similarity to successful movies released recently. We do so by computing the influence of copycat features in box-office prediction. We find a positive relationship between copycat status and movie revenue. However, this effect diminishes when the number of similar movies and the similarity of their content increase. Overall, our work develops sophisticated deep learning tools for studying the movie industry and provides valuable business insight.**

***Index Terms*—visually grounded textual representation, box-office prediction, copycat movies, content similarity, movie keywords, movie posters, model interpretability**

I. Introduction

MOVIE is a preeminent form of art in the modern era, but the business side of movie production is often less than glamorous. Statistics [1] show that box-office revenues have a long-tailed and bimodal distribution, with a small number of movies taking the lion's share of profits. The skewed distribution of movie revenues and ever-increasing production costs mean that movie production carries significant risks. The exorbitant risks underscore the importance of understanding market dynamics and accurately predicting box-office revenues [2], [3].

One risk mitigation strategy that has become increasingly popular in recent years is to produce movies similar to recent successes [4]. This strategy includes sequels, franchise movies set in the same fictional world, such as the Marvel Cinematic Universe movies and Justice League movies, and movies featuring similar story themes.

TABLE I: Examples of user-generated keywords from TMDB.

action, criminology, fbi, psycho, aircraft, robot
love, hate, high school, father-daughter relationship,
paris france, kingdom, based on novel or book

However, the failure of superhero sequels like *The Flash* (2023), which resulted in losses exceeding $200 million, may have cast doubt on the effectiveness of this strategy. Interestingly, the literature on the sequel strategy contains inconsistent findings. Some research finds that sequel movies benefit from high name recognition that attracts a large audience [5]–[7]. Others suggest that repetition of similar content can lead to audience saturation that negatively affects the box office [8], [9]. In addition, most existing studies are limited by small datasets and often analyze only the first sequel movie. Thus, there remains much room for further examination.

In this paper, we study the task of predicting movie box-office performance based on an extended definition of such similar movies — *copycat* movies that share a common story theme. We argue that the franchise/sequel strategy should be examined from a broader perspective, specifically through the lens of content similarity. For instance, following the success of *The Hunger Games* (2012), movies with similar themes, such as *The Maze Runner* (2014) and *Divergent* (2014), were quickly pushed to the market. Although these movies are not exact sequels, nor do they directly share the same fictional universe, they do share common characteristics, such as strange dystopian worlds, young adults working as a team, and strong female lead characters. Hence, we coin the term *copycat* to describe this phenomenon. Note that a sequel or franchise can be categorized as a copycat movie only if it covers similar themes in terms of story content.

We propose sophisticated machine learning techniques to overcome several methodological challenges that arise when studying copycat movies defined based on story/content themes. The first challenge is to quantify content similarity and identify copycats. We utilize user-generated movie keywords from The Movie Database (TMDB)[1] as proxies for movie content themes. Table I shows some example keywords. Compared to traditional genre categories, these keywords provide a finer content categorization, including topic, plot, emotion of the

*This article is a significantly revised and expanded version of a paper originally presented at the IEEE ICME 2023, held on July 10th in Brisbane.

[1]www.themoviedb.org

plot, and even source-related information.[2] The keyword data, which is used to define copycat movies, is highly idiosyncratic, featuring many near-synonyms and missing occurrences. To deal with these issues, we cluster the keywords using word embeddings learned from textual data and movie-keyword co-occurrence statistics, producing informative keyword clusters that allow the identification of similarly themed movies.

The second challenge is building a strong model that predicts box-office revenue accurately while accounting for the impact of the copycat strategy. To this end, we propose a multimodal pretraining strategy that grounds content keywords in the visual imagery of movie posters. Our approach is motivated by the observation that the meaning of keywords in the movie context may be subtly different from the meaning in daily usage, which is captured by pretraining on regular text corpora. For example, the movie genre *action* may be associated with explosions, car chases, or martial arts, substantially deviating from its dictionary definition. The keyword *robot* typically refers to humanoid robots in science fiction or animated movies rather than robotic arms on an assembly line. The proposed pretraining strategy incorporates movie posters into the representation learning of keywords, producing pretrained network parameters that are conducive to movie revenue prediction.

Empirical results reveal the effectiveness of the proposed technical improvements. Visual grounding pretraining reduces test error by 14.5% compared to a strong random forest baseline and by 4% compared to a pretrained BERT model with the same number of parameters. Our results are comparable to those obtained from the advanced multimodal large language model backbone, LLaVA-7B [10], and slightly outperform those achieved with the CLIP [11] backbone. On top of this strong model, we show that incorporating movie content features indeed improves the box-office revenue prediction. Specifically, we show movie content variables (e.g., keywords, copycat-related variables) significantly reduce the prediction error, achieving up to 6.9% improvement over the baseline model.

With this paper, we make the following contributions. The first three are technical contributions, whereas the last one provides business insight.

- We construct a multimodal dataset [3] of 35,794 movies, consisting of textual information and posters, to facilitate box-office revenue prediction.
- We propose a method to identify *copycat* movies that contain non-original content, using user-generated keywords from our dataset as descriptors of the movie content. To model keywords effectively, we use co-occurrence-based embeddings to summarize the long-tailed keyword list into a concise set.
- We propose a two-stage training procedure for the box-office prediction task, including a self-supervised stage that learns informative keyword representations using visual grounding and masked field prediction, and a second stage of finetuning on box-office data.
- In order to provide insight for business decision-making, we identify the key features influencing box-office revenue and quantify the effects of the *copycat* strategy. Our study indicates that the success of copycat movies depends on their relative timing with other copycat movies.

[2]More details are in the keyword contribution guide located at https://www.themoviedb.org/bible/movie

[3]https://github.com/jdsannchao/MOVIE-BOX-OFFICE-PREDICTION

II. Related Work

A. Predicting Movie Success

Previous works have attempted to predict various indicators of a movie's commercial and artistic success, including the box office [12]–[14], return on investment [15], [16], ratings [17], and awards (or nominations) [18]. More recently, with the advancement of Machine Learning, applications of deep networks in such tasks have begun to gain attention [19]–[23].

Aside from commonly adopted numeral features, available textual features include movie reviews and movie content. Previous research has mainly focused on creating topic distance matrices and conducting sentiment analysis. Studies constructing topic distance matrices have utilized the Bag-of-Words approach [15], [16], Latent Dirichlet Allocation [12], and the pretrained fastText word embeddings [23]. Sentiment analysis has also been used to analyze audience reviews, as word of mouth is an important factor for box-office prediction [13], [14], [24]. Using movie content information is crucial for other downstream tasks, such as movie question answering and recommendations. The methods used include encoding subtitles with skip-gram [25] and LSTM [26] or transforming subtitles into semantic descriptions [27], [28].

Unlike the aforementioned approaches, our approach to analyze movie content builds upon the pretrained embeddings by incorporating visual poster information through self-supervised pretraining. To our knowledge, the only prior work integrating poster information with text using deep learning models for box-office prediction is [21], which employs an evolutionary algorithm to select the optimal convolutional neural network (CNN) architecture, focusing on the layer at which visual features should be fused. In contrast, our approach does not directly integrate visual features into the box-office prediction task. Instead, we improve the pretraining of text embeddings through a visual grounding method.

B. The Copycat Effect

A copycat commonly refers to a product that heavily imitates the functionality, design, and content of existing products [29]–[31] and can be found in various industries, such as consumer goods [29], [30] and mobile apps [31]. In the movie industry, copycats can be considered as those works that closely resemble previous blockbusters in terms of content, just like the previous example of *The Divergent* (2014), which can be seen as a copycat of *The Hunger Games* (2012).

Studies on the effect of copycat products over original products on profit yield have sparked debates in various industries. Copycat mobile games, for instance, have generated higher revenues than originals due to similar pricing and more downloads [31]. However, in advertising, style imitation

makes it harder for products to stand out as competition for consumer attention intensifies [29]. In the movie industry, existing research focuses on the profitability of sequel movies [5]–[9]. However, our definition of "copycats" is broader than "sequels" and not limited to the same cast team or the same story background.

We define the concept of "copycat movies" and conduct a fine-grained analysis on a large-scale dataset. We find that copycat movies can generate higher box-office revenue in the early stages due to their similarity to blockbusters. However, over time, the worn-out effect may become more prominent and lead to audience fatigue and boredom [29], which in turn results in a decline in box-office revenue.

C. Self-supervised Multimodal Pretraining

The success of pretrained textual models such as BERT [32] has inspired a series of pretrained multimodal models [33]–[36], often leveraging the masked language modelling (MLM) objective. Similar to a denoising autoencoder, the MLM objective trains the model to predict masked portions of the input. This seemingly simple training technique has demonstrated effectiveness across various downstream applications. Another line of work, such as CLIP [37] and BLIP [38], trains the network to distinguish between correct and incorrect image-text pairs.

The symbol grounding problem [39], a classic problem in cognitive science, concerns how words can gain their meaning as pointers to other concepts and objects. Computationally grounding textual tokens in visual images has demonstrated success in some applications [40]–[45]. In this work, we use movie posters as a visual grounding source for textual tokens – keywords. A movie poster is a widely used visual medium to promote a movie long before its release. Thus, we ground the tokens using objects from a single poster, allowing multiple associations between tokens and objects. Unlike previous works that retrieve or generate relevant images for the textual descriptions, in our task, the correspondences between the keywords and the poster are not known *a priori* and must be discovered in a multi-instance manner.

III. Box-office Data Collection and Prediction Network

Here we define the problem, outline data collection and feature engineering, and present our network architecture with self-supervised pretraining.

A. Problem Definition

Accurately predicting a movie's box-office revenue y_i is a complex task influenced by multiple factors, including content, release timing, cast, and marketing. To tackle this challenge, we propose deep learning models with self-supervised training that leverage both textual and visual features to predict revenue. Formally, our goal is to learn a predictive function f that maps movie information m_i to its corresponding box-office revenue y_i.

TABLE II: Examples of the clustering results.

Cluster Label	Elements
love-hate	'love', 'loved', 'hate', 'unhappy', 'waiting', 'happy', 'grateful', 'lucky', 'expecting', 'loving'
superhero	'superhero', 'villainess', 'villain', 'symbiote', 'sidekick', 'superhuman', 'teamup', 'nemesis', 'superheroes', 'supervillain'
psychopathy	'psycho', 'psychotic', 'pyromaniac', 'psychopathic', 'homicidal', 'deranged'

B. Data Collection

To set a public benchmark for box-office prediction, we collect metadata for 35,794 movies from TMDB, spanning from 1920 to 2022. The total box-office revenue for each movie during its release period is retrieved from IMDbPro.com.

C. Keyword Content Features

The dataset includes user-generated keywords for each movie, yielding 7,700 unique keywords for 35,794 movies. We observe many rare words and near-synonyms that may hinder learning. Rare keywords lack training data, which may prevent accurate embedding learning. Synonyms and near-synonyms would force the model to learn dissimilar embeddings for words with similar meanings. We also noticed that some less popular movies have limited views, resulting in missing keywords. These inadequate keywords fail to fully reflect the content of the movie. To tackle these issues, we utilize keyword lexical similarity and co-occurrence statistics to create keyword clusters, effectively tackling both the issues of near-synonyms and missing keywords.

We use 300-dimensional embeddings computed by fastText [46] to represent the lexical information. To capture co-occurrence statistics, we employ the term-frequency inverse-document-frequency (TF-IDF) matrix and the LEPORID [47] technique, which is a regularized Laplacian Eigenmap [48]. We extract the first 50 dimensions of eigenvectors as the embedding to represent a keyword. The final representation is the 350-dimensional concatenation of the two vectors. We then perform average-link agglomerative clustering and use the resultant 1,140 keyword clusters as features of movies. We show some examples of the clustering results in Table II. We discuss the effect of keyword clusters instead of using raw keywords later in §IV and Table VIII.

D. Copycat Features

Copycat movies are driven by the desire to replicate the success of a blockbuster movie. Therefore, to identify copycat movies, we first define blockbusters as those generating at least \$10 million in revenue and having a revenue-to-budget (return on investment) ratio of at least 3. This resulted in 2,486 blockbusters in our dataset. Next, we quantify the content similarity between each movie and a blockbuster as Jaccard similarity $J(A, B) = \frac{|A \cap B|}{|A \cup B|}$ over the keyword sets A and B, where A and B denote the keyword clusters associated with the two movies, respectively.

For each blockbuster movie, we keep the top 10 most similar movies released within ten years after as its *copycats*. We keep the Jaccard similarity as a feature called *copycat similarity*. If a movie is not considered as a copycat, its value is set to zero. This feature enables us to examine the relationship between a movie's revenue and its similarity to blockbusters. Additionally, we use the chronological order of the 10 copycat movies for each blockbuster as a feature called *copycat rank*. If a movie is not a copycat, its copycat rank is zero. This feature allows us to examine the relationship between release time and the success of copycat movies.

A related concept is franchise movies. These movies are typically set in the same fictional universe, such as the Marvel Cinematic Universe or the Alien Universe, and share the same fictional universe, the common artistic vision, a compatible and co-referencing storyline, and continuous marketing strategies. While these movies may not be highly similar in content, they still benefit from the advertising and reputation of previous movies. Controlling for this feature allows us to more accurately observe the copycat effect. The label indicating whether a movie belongs to a franchise or not is sourced from the TMDB dataset, which we adopt and refer to as *franchise*. Of the 35,794 movies, we identify 4,211 copycats, 956 of which are also franchise movies. The statistics on copycats and franchises can be found in Figure 5 and Table IX in the Supplementary Material.

E. Other Common Movie Features

We also conducted feature engineering across four main movie feature categories: (1) Basic information, such as genres and MPAA ratings; (2) Production and marketing information, including the production and distribution companies as well as the budget; (3) Release timing and market competition, where competition refers to the number of other movies in the same genre released around the same time; and (4) The box office influence of the cast and crew, termed "Star Power," where we analyze the number of previously movies released by cast and crew to access their experience, and calculate the average box office revenue of their past movies to measure their profitability. The details can be found in § B and Table X of the Supplementary Material.

F. Self-supervised Pretraining

Figure 1 shows the overall pipeline. In the first stage, we pretrain a Transformer network on the MLM and visual grounding objectives. Next, we freeze the token embeddings and finetune the network on box-office prediction.

Token Embedding and Numerical Embedding. For each value in discrete features (i.e., a discrete token), we create an embedding vector whose values are learned from data. For real-valued features, we adopt prototype-based numeral embeddings [49]. Formally, the embedding function is formulated as $\mathrm{NE}(x) : \mathbb{R} \rightarrow \mathbb{R}^D$ that maps a real number x to a D-dimensional vector with the component $\mathrm{NE}_i(x) = \exp\left(-\frac{\|x-q_i\|_2}{\sigma^2}\right)$, where $\{q_i\}_{i=0}^{D-1}$ are D evenly spaced numbers over a specified interval, e.g., $[-10, 10]$. Before applying the numerical embedding function, we normalize the values using logarithm or min-max normalization, depending on whether or not the feature has a long-tail distribution.

Masked Field Prediction. We adopt a pretraining objective similar to the masked language modeling task, which has been shown to be effective for natural language understanding [32] and multimodal understanding [33]. We randomly mask one token from each group of input features—genres, keywords, director/writer names, and actor names—and train the network to predict the missing token. The prediction is formulated as cross-entropy losses, which we denote as $\mathcal{L}_{\mathrm{CE}}$. By training the network to predict missing fields, we encourage the network to learn correlations between inputs, which could mitigate data scarcity issues.

Structured Visual Grounding. Understanding keywords, which reflect movie content, is a challenging task. We propose to ground the keywords in the visual information from movie posters by contrastive learning that encourages high similarity between a poster and the corresponding content keywords and suppresses the similarity between incorrectly paired posters and keywords. First, we perform object detection on the poster with an off-the-shelf network. We denote the extracted object features from the i^{th} movie as $\mathcal{Z}_i = \{\boldsymbol{z}_m\}_{m=1}^{M}$, M refers to the number of objects. Note that we use the subscript i to denote the movie index. We also use contextualized embeddings of the keywords from the output of the Transformer network, denoted as $\mathcal{X}_i = \{\boldsymbol{x}_k\}_{k=1}^{K}$, K refers to the number of keywords.

We define the similarity between the poster and the keywords as

$$\mathrm{sim}(\mathcal{X}_i, \mathcal{Z}_i) = \sum_{(\boldsymbol{x},\boldsymbol{z})\in\mathcal{X}_i\times\mathcal{Z}_i} \exp\left(\frac{\boldsymbol{x}^\top \boldsymbol{z}}{\|\boldsymbol{x}\|_2\|\boldsymbol{z}\|_2}\right), \tag{1}$$

where $\times$ denotes the Cartesian product and $\|\cdot\|_2$ denotes the L2 norm. Due to many-to-many relations between objects on the poster and the keywords, we follow [50] to define the similarity as the sum of similarities of all possible pairs. To illustrate this, we show one example poster and the associated keywords in Figure 2. Keywords in boxes of the same color belong to the same keyword cluster (e.g., "quadriplegia" and "handicapped" belong to the red cluster). One keyword cluster can match multiple objects, and one object may be grounded in multiple clusters. For instance, the cluster "quadriplegia" is grounded by the wheelchair, the tire, and the sitting man; the sitting man is related to the red and the purple clusters.

With randomly sampled negative pairs (i', j'), we define the visual grounding (i.e., VG) loss, $\mathcal{L}_{\mathrm{VG}}$, as

$$\mathcal{L}_{\mathrm{VG}} = -\frac{1}{N}\sum_{i=1}^{N} \log\left(\frac{\mathrm{sim}(\mathcal{X}_i, \mathcal{Z}_i)}{\mathrm{sim}(\mathcal{X}_i, \mathcal{Z}_i) + \sum_{(i',j')} \mathrm{sim}(\mathcal{X}_{i'}, \mathcal{Z}_{j'})}\right) \tag{2}$$

where N is the total number of movies in the training set.

G. Finetuning on Box-office Prediction

In the finetuning stage, we train the network to predict box-office revenues. We generate the prediction by feeding the average output from all input positions into a fully connected

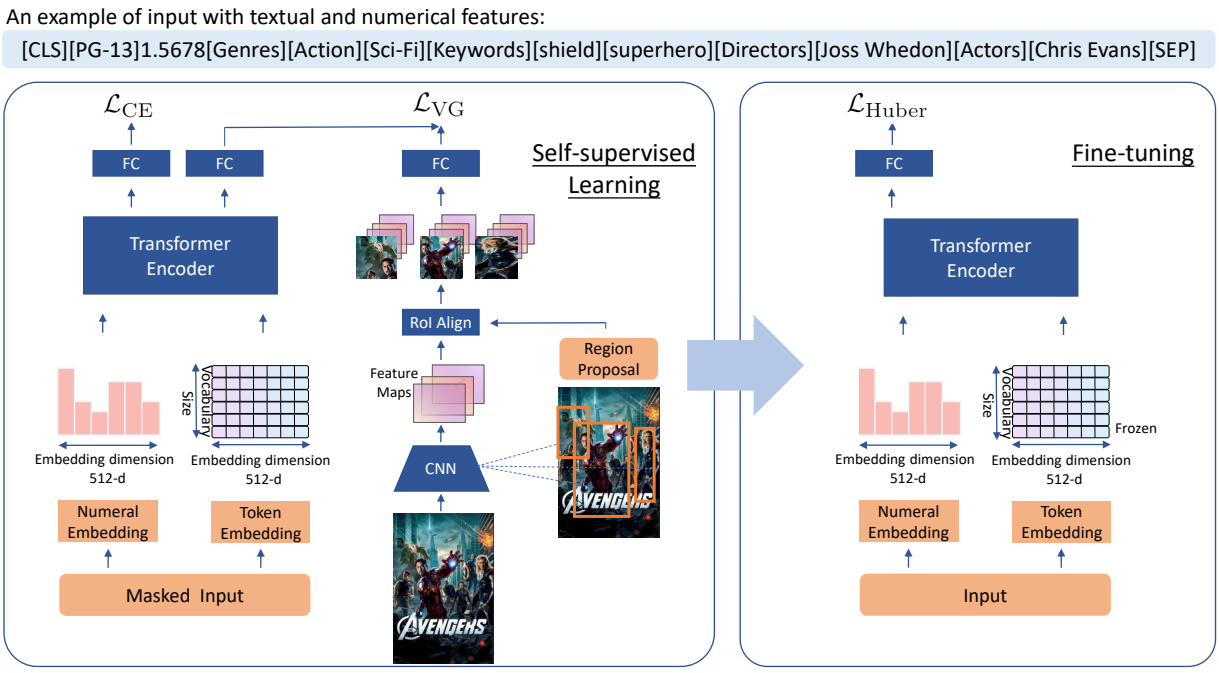


Fig. 1: The overall pipeline of self-supervised pretraining and finetuning on the box-office prediction task. The token embeddings are frozen during finetuning.

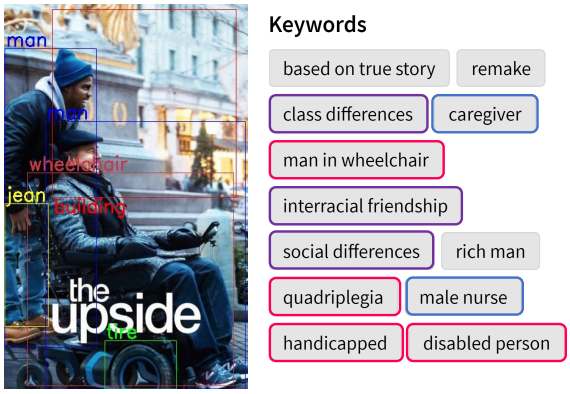


Fig. 2: Multiple objects and keywords alignments for the movie *The Upside* (2019)

layer. Since revenues follow a long-tailed distribution, we take the base-10 logarithm of the revenue as the target value. To further mitigate the impact of outliers, we train the network using the Huber loss,

$$\mathcal{L}_{\text{Huber}} = \begin{cases} 0.5\,(y-\hat{y})^2\,, & \text{if } |y-\hat{y}| < 1 \\ |y-\hat{y}| - 0.5, & \text{otherwise} \end{cases}, \tag{3}$$

where y is the ground truth and $\hat{y}$ is the prediction. Huber loss uses an absolute value when $|y-\hat{y}| > 1$ and the square of the error when $|y-\hat{y}| \leq 1$. This makes it less sensitive to outliers compared to Mean Squared Error (MSE) and it is differentiable at 0 unlike Mean Absolute Error (MAE). It is widely used in computer vision (CV) object detection as Smooth L1 loss, such as in Fast R-CNN [51].

IV. Experimental Results

A. Setup

We use stratified sampling to divide the data into train, validation, and test sets in the ratios of 70/10/20, using "franchise movie" as the label for stratification. Using the method in § III-C, we cluster 7,700 keywords into 1,414 clusters. The number of clusters is tuned on the validation set.

B. Baselines

We introduce three types of baseline models. First, we have the traditional machine learning model, Random Forest (RF). We feed all numerical features and a subset of categorical variables—those with a small number of possible classes—into RF, as one-hot encoding of all categorical features would result in excessively high dimensionality. Second, we introduce pre-trained BERT models of small. This baseline does not involve MLM or VG pretraining and we only finetune it on box-office prediction. To mimic the classic BERT input, we concatenate all input tokens into one sentence while rounding numeral features to one decimal point and then apply the BERT tokenizer. Third, we select advanced multimodal models, CLIP and LLaVA-1.5-7B [10]. We choose Long-CLIP [11] because it extends the vanilla CLIP input length to 248 tokens, suitable for accommodating our input data. For the CLIP baseline, we use only textual information as input. For the LLaVA-1.5 baseline, we conduct separate tests with and without the image poster as input. We freeze both models' parameters and extract the `[CLS]` token output from the final layer. On top of this, we train linear projector layers for box-office prediction for each model. The size of this layer remains consistent across all experiments, except for the RF as it is not applicable.

C. Implementation Details

Our model consists of a 4-layer Transformer with a dimension $d_{\text{model}} = 512$, fully connected layer dimension $d_{fc} = 512$, and attention heads $H = 4$. The architecture is the same as $\text{BERT}_{\text{small}}$. During visual grounding pretraining, we randomly select up to 6 keywords per movie and up to 20 objects per poster to compute the similarity. The feature map of each object has 2048 channels and 4×4 pixels, which is the output from VinVL [52], the off-the-shelf object detection model, after ROI Align [53] and an adaptive average pooling layer. The feature map is flattened spatially and then linearly projected to $\mathrm{R}^{d_{model}}$, where $d_{model} = 512$.

We use a batch size of 2,048 when pretraining the model under the MLM objective and reduce the size to 326 when adding the VG objective. The learning rate is 3e-4. We used Adam optimizer with a weight decay equal to 1e-4. During the finetuning stage, we freeze the model, use the last layer output of `[CLS]` token, and train a linear projector (similar to other Transformer baselines). We search for the best performance on the validation set in the combinations of learning rate in [1e-3, 3e-4, 1e-4] and batch size in [328, 512, 1024].

D. Model Performance

In Table III, we report Huber loss on the test set from the box-office prediction task for all models, as well as their performance relative to $\text{BERT}_{\text{small}}$. Pretrained BERT models easily outperform the RF but are inferior to the MLM and VG pretraining. Our best model outperformed $\text{BERT}_{\text{small}}$ by 14.5%. Notably, including VG pretraining resulted in a significant improvement on top of masked MLM. The addition of VG reduced the loss from 0.3102 to 0.3037.

Compared to other advanced multimodal baselines, first, the result shows that Long-CLIP (Huber Loss = 0.3213) underperforms compared to our proposed pre-training method (Huber loss = 0.3037). Although CLIP is pre-trained on a general multimodal corpus, our method is self-pretrained on movie context data with visual grounding enhancement. By aligning the movie content representation (keywords) with movie context (poster), our approach achieves better performance with a three times smaller parameter size.

Then, we tested LLaVA without poster input, following a similar evaluation process as with CLIP. The resulting performance was Huber Loss = 0.3092, slightly worse than our model (Huber Loss = 0.3037). We also tested LLaVA with poster input, which improved the performance to 0.3028, slightly surpassing our model. However, given the vast difference in model size, number of parameters (LLaVA: 7B vs. Ours: 161M), and pre-training data, the effectiveness of our model remains well-validated.

Furthermore, since LLaVA's training data ends in March 2023[4], while our training data ends in 2022, we collected additional box-office data from April 2023 to December 2024 to serve as a new test set (sample size = 789). On this dataset, our model outperformed LLaVA (ours: 0.7272 vs. LLaVA: 0.7406), as shown in Table IV.

[4]https://lmsys.org/blog/2023-03-30-vicuna/

TABLE III: Comparison of the test Huber loss between models. A lower loss indicates higher prediction accuracy.

Model	Test Huber Loss ↓ (% improve. over baseline)
Random Forest (RF)	0.3677 (+3.5%)
Standard $\text{BERT}_{\text{small}}$	0.3553 (baseline)
Long-CLIP init.	0.3213 (−9.6%)
LLaVA-1.5-7B w/o Poster init.	0.3092 (−13.0%)
LLaVA-1.5-7B with Poster init.	0.3028 (−14.8%)
(Ours) BERT embeddings init.	0.3137 (−11.7%)
+ MLM pretraining	0.3102 (−12.7%)
+ MLM&VG pretraining	0.3037 (−14.5%)

TABLE IV: Comparison of the test Huber loss between LLaVA and our model with VG pretraining on movies released after 2023 Mar.

Model	Test Huber Loss ↓ (% improve. over baseline)
LLaVA-1.5-7B with Poster init.	0.7406 (baseline)
BERT embeddings init. + MLM&VG pretraining	0.7272 (−1.8%)

To make the loss more interpretable based on the original scale of the box office, we calculated the Mean Absolute Percentage Error (MAPE) using the following formula:

$$\text{MAPE} = \frac{1}{N}\sum_{i=1}^{N}\left|\frac{\hat{y}_i - y_i}{y_i}\right| \tag{4}$$

We then divided the movies into four buckets based on their total box-office revenue. Table V shows that our predictions are more accurate for high box-office movies in general. For a movie with an actual box-office revenue of 1 billion (1B), a MAPE of 0.52 indicates that the predicted revenue typically falls between 0.48 billion and 1.52 billion, demonstrating a good level of prediction accuracy.

E. Ablation Study

This subsection discusses how different settings can improve the model performance. The keywords used in models have been clustered using the clustering method introduced in §III-C otherwise specified.

Effects of Content-related Variables. We examine the impact of copycat variables and keyword variables on box-office revenue prediction. Table VI shows that incorporating copycat-related variables leads to an average improvement of 2%. Table VII shows that adding content keywords leads to an average performance improvement of 6.4%. The table does not contain the VG loss, as it is impossible to apply the VG loss without content keywords.

Effects of Keyword Clusters. We verify the advantage of keyword clusters over raw keywords. Table VIII compares models trained with keyword clusters ("Clustering") with those with raw keywords ("Keywords"). In most cases, keyword

TABLE V: MAPE on box-office revenue after reversing the log10 transformation. Movies are categorized into four revenue buckets.

Revenue Bucket	MAPE ↓	Movie Count
<$1M	46.582	3,698
$1M-$100M	0.934	3,161
$100M-$1B	0.571	326
>$1B	0.522	11

TABLE VI: Comparison between the same model before and after incorporating copycat-related variables.

Model	Copycat-related Variables	Test Huber Loss ↓ (% performance difference)
BERT embeddings init.	✗	0.3207
	✓	0.3137 (−2.2%)
+ MLM pretraining	✗	0.3163
	✓	0.3102 (−1.9%)
+ MLM&VG pretraining	✗	0.3094
	✓	0.3037 (−1.8%)

clusters provide performance gains, especially when pretrained BERT embeddings are used. One possible reason is that near-synonyms have similar BERT embeddings, making them difficult for model to differentiate, and clustering can alleviate this problem.

Additionally, the keyword clustering provides greater benefit in VG pretraining models. The contrastive learning in VG pushes the embeddings of different keywords, $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$, apart, which can be problematic if the two were in fact synonyms. In the keyword clustering setting, most synonyms are already clustered together, thereby mitigates this issue and enhances the discriminative power of contrastive learning.

V. Interpreting Feature Importance

Identifying variables that influence box office may facilitate the decision-making by movie producers and investors, such as deciding whether to produce original content or imitate blockbuster movies. In this section, we apply two interpretability methods, Attention Rollout [54] and LIME [55], to estimate the impact of the features curated in §III.

A. Attention Weights, Gradients and Attention Rollout

We analyze the internal states of the model, specifically the attention weights, to assess the influence of a variable on the model's predictions. The attention weights reflect how the information of the tokens flows from input to output in the model – tokens with higher attention weights contribute more to the representation of the tokens in the next layer. Thus, attention weights are used to provide plausible and meaningful interpretations [56]–[58].

We adopt the technique Attention Rollout [54], [59]. Consider a Transformer model with L layers. For the attention matrix A_l at layer l, the weights in the m-th column indicate how much each token attends to the m-th token during token prediction. The weights in the n-th row show how much the n-th token attends to other tokens during the generation of the n-th token. The information flow from the m-th token at layer l - 1 to the n-th token at layer l is calculated by the dot product between the m-th column of $A_{l\text{-}1}$ and the n-th row of A_l. This operation sums the attention weights across all possible paths between the two tokens. Therefore, by performing matrix multiplication $A_l A_{l\text{-}1}$, the resulting matrix represents the amount of information flowing to each token from every token in the previous layer. Notably, we choose to elementwise multiply $(*)$ the original attention matrix by its gradients G_l with respect to the prediction loss, to highlight the attention weights that are important for minimizing the prediction loss. All negative values are then set to zero. The details are shown in Equation 5.

$$
\begin{aligned}
A'_l &= \max(G_l, 0) * A_l \\
\tilde{A}_l &= \begin{cases} A'_l & \text{if } l = 1 \\ A'_l \tilde{A}_{l\text{-}1} & \text{if } l > 1 \end{cases}
\end{aligned} \tag{5}
$$

The n-th row of $\tilde{A}_L$ at the last layer represents the attention weights of each token, propagated from the input layer to the n-th token in the output layer. We are only interested in the information flowed into the `[CLS]` token, as `[CLS]` token is the sole token used to predict the box-office. Thus, the first row $\vec{a} = \tilde{A}_L[0, :]$, which represents the influence of all tokens on the `[CLS]` token, is retained as the final result.

We next aggregate the vector $\vec{a}$ by averaging over the test set

TABLE VII: Comparison between the same model before and after incorporating content variables.

Model	Copycat Var. and Keywords	Test Huber Loss ↓ (% performance difference)
BERT embeddings init.	✗	0.3370
	✓	0.3137 (−6.9%)
+ MLM pretraining	✗	0.3298
	✓	0.3102 (−5.9%)

TABLE VIII: The comparison between the same model before and after incorporating keywords clustering.

Model	Keywords Clustering	Test Huber Loss ↓ (% performance difference)
Random embeddings init.	✗	0.3265
	✓	0.3290 (+0.8%)
+ MLM pretraining	✗	0.3133
	✓	0.3109 (−0.8%)
+ MLM&VG pretraining	✗	0.3109
	✓	0.3070 (−1.3%)
BERT embeddings init.	✗	0.3249
	✓	0.3137 (−3.4%)
+ MLM pretraining	✗	0.3226
	✓	0.3102 (−3.8%)
+ MLM&VG pretraining	✗	0.3182
	✓	0.3037 (−4.6%)

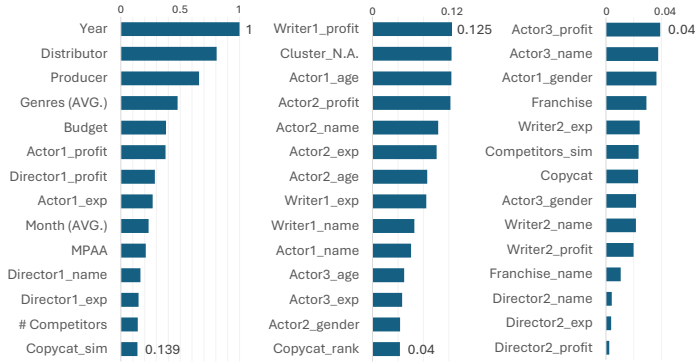


Fig. 3: Attention rollout ranking. The values are normalized so that the maximum variable, `Year`, has a value of 1. The '1/2/3' designation for actors, directors, or writers indicates the order of prominence or billing.

based on the occurrences of each variable value. For Genre and Month, we track the attention rollout values for each element value.

Figure 3 shows the attention rollout ranking which is divided into three tiers from the highest to the lowest. The top-ranked features that have the greatest impact on box-office revenue are the movie's `Year`, `Distributor`, `Producer`, `Genres` (on average) and `Budget`.

Among the copycat-related features, the `copycat_sim` is highly ranked. This suggests that the degree of a given movie's content being similar to other blockbuster movies is important in box-office prediction.

While attention rollout shows the variable importance, it does not indicate whether each variable's impact is positive or negative. Furthermore, it does not quantify the magnitude of each variable's impact on the output (target). An additional interpretability technique is required to provide further insights.

B. LIME

The LIME (Local Interpretable Model-Agnostic Explanations) method [55] provides feature importance for a given data point. Specifically, we perform the following steps: Given the input data point, a K dimensional vector, LIME first performs N (here $N = 5000$) random perturbations over all the dimensions, yielding N synthetic data points, denoted as X_s. The perturbations follow the same distribution as the training data. Next, we gather model predictions on each synthetic data point, resulting in N pseudo-labels, $\hat{y}_s \in \mathbb{R}^{N\times 1}$. Finally, LIME fits a Lasso regression model on $\langle X_s, \hat{y}_s \rangle$, and the K regression coefficients represent the feature importance in the local model around the given input data point. After that, feature importance can be aggregated over many input data points.

Due to the sequential and heterogeneous nature of our data, which include both numerical variables and textual tokens, we perform the feature perturbation before embedding the input sequence. We perturb only a subset of features, including i) all numerical variables, and ii) most of the textual variables that have been simplified into categorical variables with no more than 12 categories. We do not perturb the extensive list of cast names and keywords. While these features are to the Transformer model, they are excluded from the Lasso regression, and hence will not produce LIME coefficients.

One thing to note is that since LIME provides a local explanation for each data point, for categorical variables, the LIME coefficient represents the influence of the specific variable value relative to other possible values. For instance, the "Distributor" variable is categorized into 6 classes and we obtain 6 different LIME coefficients, one for each class.

C. Results and Discussion

Figure 4 displays the impact of variables as reflected by the LIME coefficients.

Importance of Commonly Used Features. Figure 4(a) illustrates the top 7 most important features, as reflected by the largest absolute LIME coefficients. Except for `Budget` and `Actor1_exp`, the rest are categorical features. Similar to attention rollout, LIME also identifies the `Producer` and `Distributor` features to have significant impacts. As categorical features can take on different values, Figure 4(a) reflects the aggregated impact of each feature by averaging the LIME score for every possible values. We then show the impacts of those values in Figure 4(b)-(f).

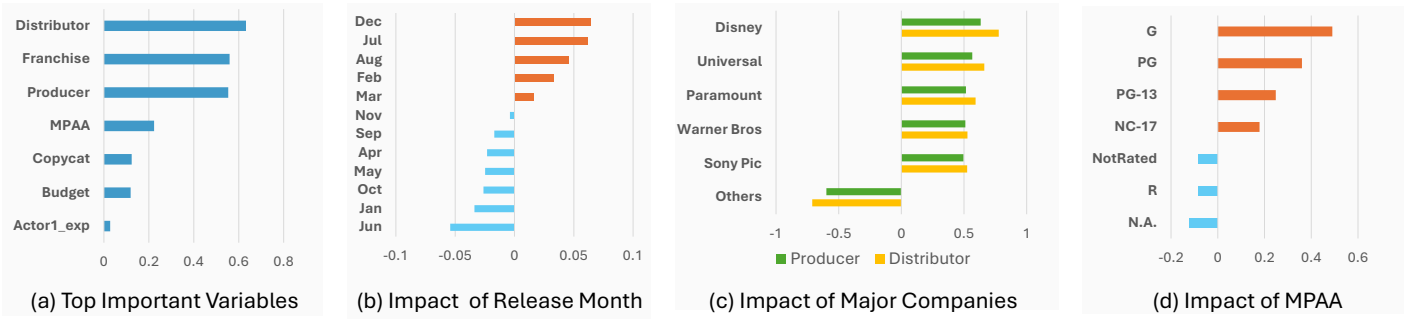
Next, we see a wide variation within `release_month` in Figure 4(b). The coefficient -0.058 for June indicates that movies released in June are expected to have a lower box-office performance. In contrast, movies released in December are expected to see the increase in the box-office performance.

Moreover, the marketing efforts by major movie companies have huge impacts on box office, as illustrated by Figure 4(c). These companies enjoy strong production and distribution capabilities. For example, being produced by Disney increases revenue by 0.6 on average. The remaining four major companies are similar—a movie not produced or promoted by these companies will suffer revenue loss with coefficient of about -0.6.

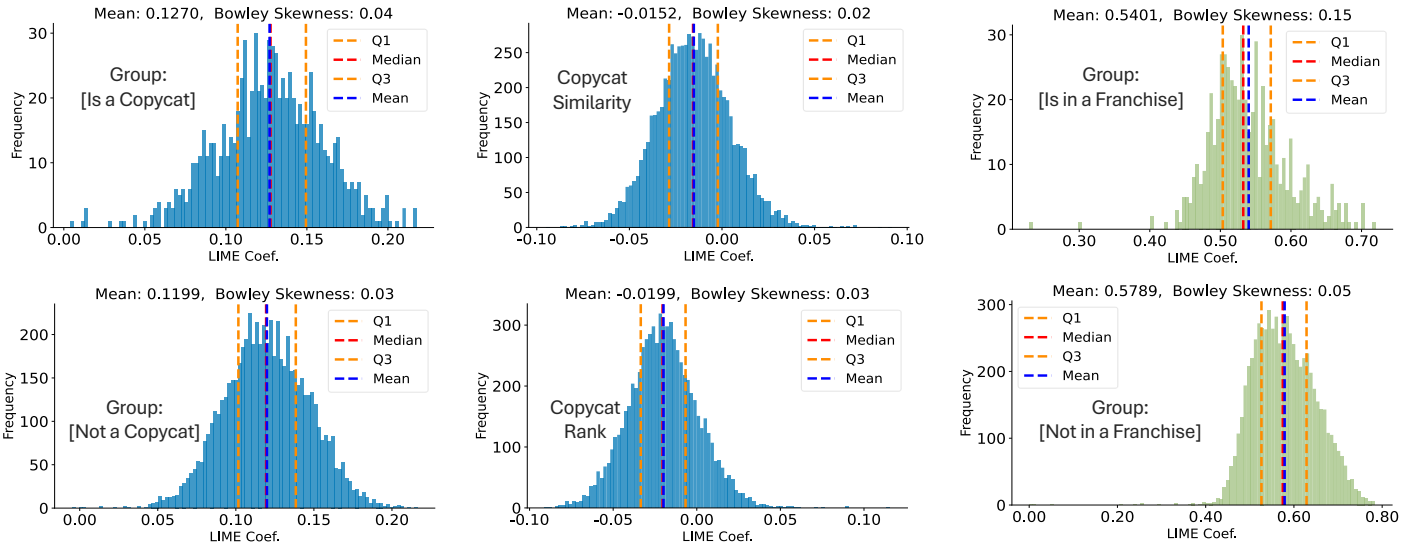
Lastly, we have information about MPAA rating in Figure 4(d). As expected, a G rating shows the highest positive impact while an R rating and being unrated lower the revenue. Interestingly, NC-17 with sexual content turns out to have a positive impact.

Importance of Copycat Features. As shown in Figure 4(e) Top Left, the average LIME coefficient for the *copycat* variable among the copycat movies is 0.1270. Meanwhile, the LIME coefficient for changing a non-copycat movie to a copycat movie, other features being equal, is 0.1199. Combining the above findings, being a copycat is associated with an approximately 0.12 increase in $\log_{10}$ box-office revenue or roughly $318,256 for a movie with revenue of 1 million dollars.

Interestingly, as the LIME coefficient for *copycat similarity* variable is negative ($\beta = -0.0152$, as shown in Figure 4(e) Top Right), we conclude that an increase of content similarity leads to a decrease in box-office revenue. Likewise, an increase in the *copycat rank*, i.e., the more copycat movies stemming from the same blockbuster are released before the focal movie, has a negative impact on box-office revenue ($\beta = -0.0199$, Figure 5(e) Bottom Right).



(a) Top Important Variables (b) Impact of Release Month (c) Impact of Major Companies (d) Impact of MPAA



(e) Left: Impact of the Copycat Strategy on Different Movie Groups. Right: Impact of Copycat Similarity and Rank.

(f) : Impact of the Franchise Strategy on Different Movie Groups.

Fig. 4: The feature impacts using the LIME method. (a)-(d): Results for non-content related variables at the aggregation level. (e)-(f): The distribution plot of LIME coefficients for copycat-related variables (e) and the franchise variable (f).

Taken together, we demonstrate an important insight: In general, movies categorized as copycat produce higher box-office revenue than original, non-copycat movies. However, too much similarity to the original blockbuster movie or the repetition of the same content over time across different copycat movies suppresses profitability.

The Franchise Feature. Next, as shown in Figure 4(f) Top and Bottom, the LIME model consistently estimates an average positive impact of around $0.54 \sim 0.57$ on earnings for movies that are part of a franchise, whether they are originally franchise movies or non-franchise movies under LIME perturbation. Additionally, we find that LIME coefficients for being franchises have a large positive skew (under the Bowley skewness [60] measurement), suggesting that franchise movies are likely to have extremely high box-office performances.

Consistency between LIME and Attention. By calculating the rank correlation (i.e., Spearman's Coefficient) between the LIME and the attention rollout results, we get a value of $\rho = 0.6948$ (based on features in Figure 3, excluding textual information such as the names of casts). This indicates that the two sets of results are highly correlated with each other, suggesting our findings are not heavily reliant on the choice of the interpretability technique.

VI. Conclusion

To conclude, our work has established a comprehensive process to predict movie box-office performance and provide valuable business insights for content management. The process begins with data collection, combining tokens, numbers, and visual information to represent the data. It then proceeds to the prediction task using the Transformer model through pre-training and self-supervision, and finally interprets the model's output. We believe our findings not only shed light on the importance of multimodal elements in box-office prediction but also demonstrate the interpretability of deep models in predictive tasks. This comprehensive approach offers significant implication for future research in this domain.

VII. Acknowledgments

This research is supported, in part, by the RIE2025 Industry Alignment Fund – Industry Collaboration Projects (IAF-ICP) (Award I2301E0026), administered by A*STAR, as well as supported by Alibaba Group and NTU Singapore through Alibaba-NTU Global e-Sustainability CorpLab (ANGEL). The research is also partially funded by the Nanyang Associate Professorship and National Research Foundation Fellowship (NRFF13-2021-0006), Singapore.

REFERENCES

[1] R. K. Pan and S. Sinha, "The statistical laws of popularity: universal properties of the box-office dynamics of motion pictures." *New Journal of Physics*, 2010.

[2] J. McKenzie, "The economics of movies (revisited): A survey of recent literature," *Journal of Economic Surveys*, vol. 37, no. 2, pp. 480–525, 2023.

[3] R. Behrens, N. Z. Foutz, M. Franklin, J. Funk, F. Gutierrez-Navratil, J. Hofmann, and U. Leibfried, "Leveraging analytics to produce compelling and profitable film content," *Journal of Cultural Economics*, vol. 45, pp. 171–211, 2021.

[4] J. Eliashberg, A. Elberse, and M. A. Leenders, "The motion picture industry: Critical issues in practice, current research, and new research directions," *Marketing science*, vol. 25, no. 6, pp. 638–661, 2006.

[5] S. A. Ravid and S. Basuroy, "Managerial objectives, the r-rating puzzle, and the production of violent films," *The Journal of Business*, vol. 77, no. S2, pp. S155–S192, 2004.

[6] S. Basuroy, K. K. Desai, and D. Talukdar, "An empirical investigation of signaling in the motion picture industry," *Journal of marketing research*, vol. 43, no. 2, pp. 287–295, 2006.

[7] B. Belvaux and R. Mencarelli, "Prevision model and empirical test of box office results for sequels," *Journal of Business Research*, vol. 130, pp. 38–48, 2021.

[8] T. Dhar, G. Sun, and C. B. Weinberg, "The long-term box office performance of sequel movies," *Marketing Letters*, vol. 23, pp. 13–29, 2012.

[9] S. Sood and X. Drèze, "Brand extensions of experiential goods: Movie sequel evaluations," *Journal of Consumer Research*, vol. 33, no. 3, pp. 352–360, 2006.

[10] H. Liu, C. Li, Q. Wu, and Y. J. Lee, "Visual instruction tuning," *Advances in neural information processing systems*, vol. 36, 2024.

[11] B. Zhang, P. Zhang, X. Dong, Y. Zang, and J. Wang, "Long-clip: Unlocking the long-text capability of clip," in *European Conference on Computer Vision*. Springer, 2024, pp. 310–325.

[12] E. Kim, M. Ding, X. Wang, and S. Lu, "Does topic consistency matter? a study of critic and user reviews in the movie industry," *Journal of Marketing*, vol. 87, no. 3, pp. 428–450, 2023.

[13] K. R. Apala, M. Jose, S. Motnam, C.-C. Chan, K. J. Liszka, and F. de Gregorio, "Prediction of movies box office performance using social media," in *ASONAM*. IEEE, 2013.

[14] M. Hur, P. Kang, and S. Cho, "Box-office forecasting based on sentiments of movie reviews and independent subspace method," *Information Sciences*, vol. 372, pp. 608–624, 2016.

[15] J. Eliashberg, S. K. Hui, and Z. J. Zhang, "Assessing box office performance using movie scripts: A kernel-based approach," *IEEE Trans. Knowl. Data Eng.*, 2014.

[16] M. T. Lash and K. Zhao, "Early predictions of movie success: The who, what, and when of profitability," *J Manag Inf Syst*, 2016.

[17] B. Cizmeci and Ş. G. Ögüdücü, "Predicting imdb ratings of pre-release movies with factorization machines using social media," in *2018 3rd IEEE UBMK*, 2018.

[18] P. Boccardelli, F. Brunetta, F. Vicentini *et al.*, "What is critical to success in the movie industry? a study on key success factors in the italian motion picture industry," 2008.

[19] N. Quader, M. O. Gani, and D. Chaki, "Performance evaluation of seven machine learning classification techniques for movie box office success prediction," in *IEEE EICT*, 2017.

[20] E. A. Antipov and E. B. Pokryshevskaya, "Are box office revenues equally unpredictable for all movies? evidence from a random forest-based model," *J. Revenue Pricing Manag.*, 2017.

[21] Y. Zhou and G. G. Yen, "Evolving deep neural networks for movie box-office revenues prediction," in *2018 IEEE CEC*, 2018.

[22] Y. J. Kim, Y. G. Cheong, and J. H. Lee, "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.

[23] M. Shafaei, A. P. Lopez-Monroy, and T. Solorio, "Exploiting textual, visual, and product features for predicting the likeability of movies," in *The Thirty-Second Intl. Flairs Conference*, 2019.

[24] X. Yu, Y. Liu, X. Huang, and A. An, "Mining online reviews for predicting sales performance: A case study in the movie domain," *IEEE Transactions on Knowledge and Data engineering*, vol. 24, no. 4, pp. 720–734, 2010.

[25] Z. Yuan, S. Sun, L. Duan, C. Li, X. Wu, and C. Xu, "Adversarial multimodal network for movie story question answering," *IEEE Transactions on Multimedia*, vol. 23, pp. 1744–1756, 2020.

[26] Z. Zhao, Q. Yang, H. Lu, T. Weninger, D. Cai, X. He, and Y. Zhuang, "Social-aware movie recommendation via multimodal network learning," *IEEE Transactions on Multimedia*, vol. 20, no. 2, pp. 430–440, 2017.

[27] C. Liang, C. Xu, J. Cheng, W. Min, and H. Lu, "Script-to-movie: a computational framework for story movie composition," *IEEE transactions on multimedia*, vol. 15, no. 2, pp. 401–414, 2012.

[28] K. Kurzhals, M. John, F. Heimerl, P. Kuznecov, and D. Weiskopf, "Visual movie analytics," *IEEE Transactions on Multimedia*, vol. 18, no. 11, pp. 2149–2160, 2016.

[29] P. A. Naik, M. K. Mantrala, and A. G. Sawyer, "Planning media schedules in the presence of dynamic advertising quality," *Marketing science*, vol. 17, no. 3, pp. 214–235, 1998.

[30] F. Van Horen and R. Pieters, "Consumer evaluation of copycat brands: The effect of imitation type," *International Journal of Research in Marketing*, vol. 29, no. 3, pp. 246–255, 2012.

[31] Q. Wang, B. Li, and P. V. Singh, "Copycats vs. original mobile apps: A machine learning copycat-detection method and empirical analysis," *Information Systems Research*, vol. 29, no. 2, pp. 273–291, 2018.

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

[33] J. Lu, D. Batra, D. Parikh, and S. Lee, "Vilbert: Pretraining task-agnostic visiolinguistic representations for vision-and-language tasks," *NeurIPS*, 2019.

[34] W. Su, X. Zhu, Y. Cao, B. Li, L. Lu, F. Wei, and J. Dai, "Vl-bert: Pre-training of generic visual-linguistic representations," *arXiv:1908.08530*, 2019.

[35] H. Tan and M. Bansal, "Lxmert: Learning cross-modality encoder representations from transformers," in *EMNLP-IJCNLP*, 2019.

[36] Z. Huang, Z. Zeng, B. Liu, D. Fu, and J. Fu, "Pixel-bert: Aligning image pixels with text by deep multi-modal transformers," *arXiv:2004.00849*, 2020.

[37] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark *et al.*, "Learning transferable visual models from natural language supervision," in *ICML*, 2021.

[38] J. Li, D. Li, C. Xiong, and S. Hoi, "Blip: Bootstrapping language-image pre-training for unified vision-language understanding and generation," *arXiv:2201.12086*, 2022.

[39] S. Harnad, "The symbol grounding problem," *Physica D: Nonlinear Phenomena*, vol. 42, no. 1-3, pp. 335–346, jun 1990. [Online]. Available: https://doi.org/10.1016%2F0167-2789%2890%2990087-6

[40] J. Kiros, W. Chan, and G. Hinton, "Illustrative language understanding: Large-scale visual grounding with image search," in *ACL*, 2018. [Online]. Available: https://aclanthology.org/P18-1085

[41] H. Tan and M. Bansal, "Vokenization: Improving language understanding with contextualized, visual-grounded supervision," in *EMNLP*, Nov. 2020. [Online]. Available: https://aclanthology.org/2020.emnlp-main.162

[42] L. Zhang, Q. Chen, J. Siebert, and B. Tang, "Semi-supervised visual feature integration for language models through sentence visualization," in *ICMI*, 2021. [Online]. Available: https://doi.org/10.1145/3462244.3479965

[43] Y. Lu, W. Zhu, X. E. Wang, M. Eckstein, and W. Y. Wang, "Imagination-augmented natural language understanding," in *NAACL*, 2022.

[44] Y. Yang, W. Yao, H. Zhang, X. Wang, D. Yu, and J. Chen, "Z-lavi: Zero-shot language solver fueled by visual imagination," *arXiv:2210.12261*, 2022.

[45] X. Liu, D. Yin, Y. Feng, and D. Zhao, "Things not written in text: Exploring spatial commonsense from visual signals," in *ACL*, May 2022.

[46] P. Bojanowski, E. Grave, A. Joulin, and T. Mikolov, "Enriching word vectors with subword information," *arXiv preprint arXiv:1607.04606*, 2016.

[47] Y. Zhang, B. Li, Y. Liu, H. Wang, and C. Miao, "Initialization matters: regularizing manifold-informed initialization for neural recommendation systems," in *Proceedings of the 27th ACM SIGKDD*, 2021.

[48] U. von Luxburg, "A tutorial on spectral clustering," *Statistics and Computing*, vol. 17, pp. 395–416, 2007. [Online]. Available: https://api.semanticscholar.org/CorpusID:3264198

[49] Z. Jin, X. Jiang, X. Wang, Q. Liu, Y. Wang, X. Ren, and H. Qu, "Numgpt: Improving numeracy ability of generative pre-trained models," *arXiv:2109.03137*, 2021.

[50] A. Miech, J.-B. Alayrac, L. Smaira, I. Laptev, J. Sivic, and A. Zisserman, "End-to-end learning of visual representations from uncurated instructional videos," in *CVPR*, 2020.

[51] R. Girshick, "Fast r-cnn," *arXiv preprint arXiv:1504.08083*, 2015.

[52] P. Zhang, X. Li, X. Hu, J. Yang, L. Zhang, L. Wang, Y. Choi, and J. Gao, "Vinvl: Revisiting visual representations in vision-language models," in *CVPR*, 2021.
[53] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask r-cnn," in *ICCV*, 2017.
[54] S. Abnar and W. Zuidema, "Quantifying attention flow in transformers," *arXiv preprint arXiv:2005.00928*, 2020.
[55] M. T. Ribeiro, S. Singh, and C. Guestrin, ""why should I trust you?": Explaining the predictions of any classifier," in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco, CA, USA, August 13-17, 2016*, 2016, pp. 1135–1144.
[56] S. Wiegreffe and Y. Pinter, "Attention is not not explanation," *arXiv preprint arXiv:1908.04626*, 2019.
[57] S. Vashishth, S. Upadhyay, G. S. Tomar, and M. Faruqui, "Attention interpretability across nlp tasks," *arXiv preprint arXiv:1909.11218*, 2019.
[58] J. Vig, "Visualizing attention in transformer-based language representation models," *arXiv preprint arXiv:1904.02679*, 2019.
[59] J. Gildenblat, "Explainability for vision transformers," https://github.com/jacobgil/vit-explain?tab=readme-ov-file, 2022.
[60] J. Kenney and E. Keeping, *Mathematics of Statistics*. D. Van Nostrand Company, Princeton, 1962, vol. 1.

Chao Qin is a Ph.D. student at the College of Computing and Data Science (CCDS), Nanyang Technological University (NTU) in Singapore. She is exploring how stories come to life through words, images, and media. Her passion is uncovering the hidden meanings in these diverse storytelling forms.

Kim Eunsoo is an Associate Professor of International Business at the University of Seoul. She holds a Ph.D. degree in Quantitative Marketing from the Stephen M. Ross School of Business at the University of Michigan, Ann Arbor. She was affiliated with NTU as an Assistant Professor of Marketing at Nanyang Business School. Her research focuses on quantitative modeling of user/firm-generated content and social influence, using econometric models, deep learning, and image/text analysis.

Li Boyang is a Nanyang Associate Professor at the College of Computing and Data Science, NTU. His research interests include computational narrative intelligence, multimodal learning, machine learning and data-centric AI. In 2021, he received the National Research Foundation Fellowship, a prestigious research award of 2.5M Singapore Dollars. He was a Senior Scientist at Baidu Research USA and a Research Scientist and Group Leader at Disney Research Pittsburgh. He received his Ph.D. degree from the Georgia Institute of Technology.