Actor Network Analysis & Collaboration Prediction

Social Media Analysis Term Project

Jih-Ming	Cheng-Yu	Shang-Ching	Po-Yen	Chia-Shan
Bai	Kuan	Su	Chu	Li
R11725041	R12725028	B10704096	B10704031	B10704051
@ntu.edu.tw	@ntu.edu.tw	@ntu.edu.tw	@ntu.edu.tw	@ntu.edu.tw

1 Abstract

In the film industry, selecting the right cast is often a crucial factor influencing a movie's success. Two actors who are well-matched can enhance each other's performance, creating chemistry that drives box office success and positive reviews, ultimately generating a virtuous cycle for the film.

There are generally three common types of actor collaborations. The first type is derived from the same series of characters, such as Robert Downey Jr. and Chris Evans in the Marvel Cinematic Universe, playing Iron Man and Captain America, respectively, which brought tremendous success to the series. The second type involves the same pair of actors trying out different genres in multiple films, such as Ben Stiller and Owen Wilson, who have collaborated in nine classic movies like "Meet the Parents" and "Night at the Museum," among others. These films vary in genre and are not part of the same series. The final type is the most common scenario—new actor combinations and new attempts.

However, finding suitable actors to collaborate is a complex issue. Producers need to consider the actors' past collaborations, images, personalities, and other factors to determine whether a pairing has potential. This information is often scattered and incomplete, and there may be unobservable factors affecting the influence and preferences regarding actor combinations and box office performance due to subjective judgments.

Besides these characteristics, audience reviews on movie review platforms and streaming services like Netflix can provide valuable data for analyzing the effectiveness of actor collaborations and audience reactions. Through these platforms' review data, producers can gain more feedback on specific actor pairings, thereby more accu-

rately predicting the success probability of future collaborations.

Therefore, this study aims to analyze the collaboration data between movie actors using social network analysis methods, constructing a network of these actors' collaborative relationships, and predicting potential actor combinations and types. This will optimize the pairing process and efficiency of movie production, promoting the development of the film industry. Furthermore, if user data from streaming platforms can be incorporated into the model, this method also has the potential to serve as a recommendation system for film and television works.

2 Literature Review

2.1 Starring and Movie Success

Actors have a significant impact on box office performance (Nelson et al., 2012), playing a crucial role in the film industry. Moreover, recent studies have explored the impact of co-starring on film success. A study by Kwan and Scheepers (2022) found that the collective celebrity status of a film's cast has a statistically significant, positive correlation with commercial success and public perception. These studies highlight the complex interplay between co-starring actors and film success, underscoring the importance of considering various factors in predicting box office performance.

2.2 Link Prediction Paradigms

Link prediction is a crucial task in graph machine learning with a range of applications, such as recommender systems. The existing approaches to link prediction can be divided into three main categories: path-based methods, embedding methods, and graph neural networks (GNNs) (Zhu et al., 2021; Chamberlain et al., 2022). Path-based methods, or heuristics, focus on calculating node similarities based on the paths within a graph. Early techniques in this category include the Katz index, personalized PageRank, and graph distance. More advanced metrics like PathSim and Path Ranking extend these approaches to heterogeneous and knowledge graphs. Other works including rule mining methods and path representation methods aim to search for useful and comprehensible paths within the graph. Embedding methods aim to learn representations for nodes and edges while preserving the graph's structural information. Techniques such as DeepWalk (Perozzi et al., 2014) and LINE (Tang et al., 2015) have successfully learned embeddings that show promising results in link prediction tasks and can scale efficiently to large graphs. Additionally, node2vec (Grover et al., 2016), struc2vec (Ribeiro et al., 2017), and metapath2vec (Dong et al., 2017) are popular graph embedding techniques that use random walks to generate sequences of nodes and then learn node representations. Graph neural networks have become popular for their ability to encode topological structures. In link prediction, GNNs are used to encode node representations and decode edges. Some frameworks, like SEAL (Zhang et al., 2018) and GraIL (Teru et al., 2020), explicitly encode subgraphs for each node pair, while others use autoencoder formulations. It has been shown that learning from local subgraphs outperforms traditional heuristic methods, latent feature

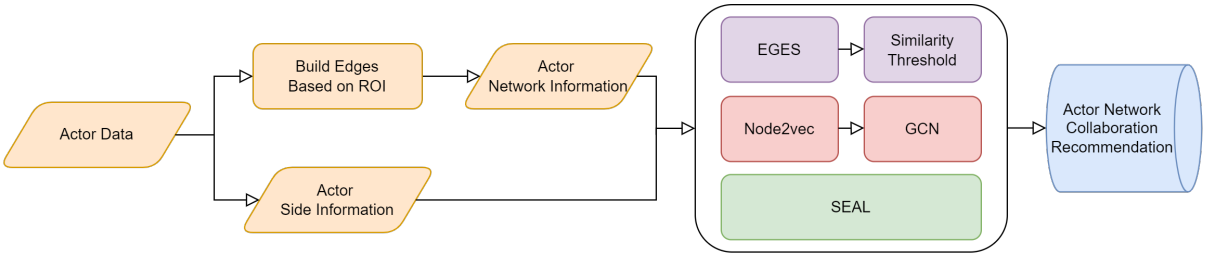

Figure 1: Actor Network Collaboration Recommendation Flowchart

methods, and network embedding methods. Despite scalability challenges with large graphs, the robustness across different graph structures have made it a prominent approach in link prediction.

2.3 Features for Link Prediction

Graph structure features, latent features, and explicit features are the three main types of features used for link prediction in networks (Zhang et al., 2018). Graph structure features are based on the observed node and edge structures of the network, such as link prediction heuristics and node centrality scores. These features can be directly computed from the network topology. While effective, they only capture a limited set of structural patterns and lack the ability to express more general patterns underlying different networks. Latent features are low-dimensional embeddings of nodes generated by factorizing matrices derived from the network, such as the adjacency matrix or Laplacian matrix. These embeddings emphasize global properties and long-range effects but fail to capture structural similarities between nodes. Additionally, latent features are transductive, meaning they cannot be applied to new nodes or networks without retraining. Explicit features represent side information about the network other than its structure, such as user profile information in social networks. These features provide additional context that can enhance the understanding of node relationships. For example, in a social network, explicit features might include demographic data, interests, or user activity patterns. By incorporating this supplementary information into the analysis, explicit features enable more nuanced predictions and insights.

3 Data Description

3.1 Dataset

The dataset utilized in this study comprises real-world data sourced from three primary files: `movie_data.csv`, `movie_budget.csv`, and `actor_data.csv`. The `movie_data.csv` file contains information about movies obtained from Kaggle's TMDB 15000 Movies Dataset (with credits), while the `actor_data.csv` file contains data about actors sourced from Data.World. Additionally, supplementary data on box office gross and budget information for movies were collected from `The Numbers` website and stored in the `movie_budget.csv` file. These datasets collectively provide a rich background for predicting collaborations between movie actors.

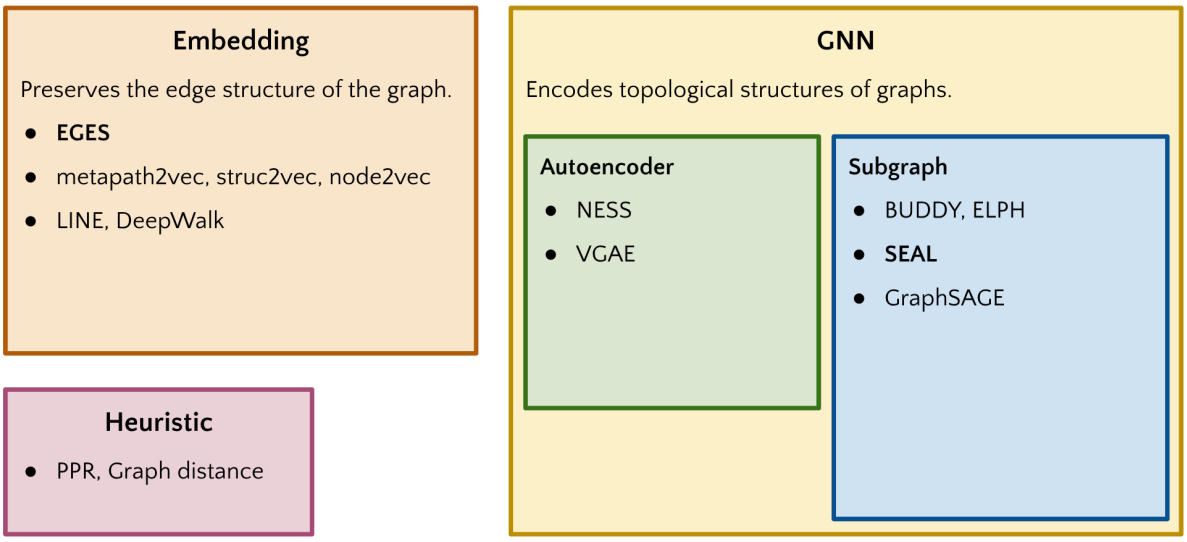

Figure 2: Link Prediction Paradigms

3.1.1 Data Format

`movie_data.csv` The columns and their respective descriptions are as follows:

- `title`: Movie title
- `release_date`: Release date of the movie
- `cast`: List of actors in the movie, with an index `order` indicating the characters' order in the movie

`actor_data.csv` The fields include:

- `name`: Actor name
- `gender`: Gender
- `popularity`: Actor's popularity
- `Date of Birth`: Date of birth
- `Birth Country`: Birth country
- `Height (Inches)`: Height (in inches)
- `Ethnicity`: Ethnicity
- `NetWorth`: Net worth
- `Age`: Age

`movie_budget.csv` The fields include:

- `title`: Movie title
- `release_date`: Release date of the movie
- `worldwide_gross`: Total box office gross
- `budget`: Movie budget

3.1.2 Data Volume

`movie_data.csv`

- Total number of movies: 2686
- Time span: January 2000, to May 2023

`actor_data.csv`

- Total number of actors: 1942

3.2 Data Preprocessing

3.2.1 Filtering and ROI Calculation

In the data preprocessing phase, the following steps were undertaken to prepare the dataset for analysis:

1. **Filtering movies released after 2000**: Only movies released after the year 2000 were retained for further analysis. This step was performed based on the `release_date` field.

2. **Merging gross and budget data**: Box office gross and budget information for movies were collected from The Numbers website and merged with the main dataset. Matching was performed based on the movie titles.

3. **Calculation of ROI (Return on Investment) for each movie**: ROI was calculated using the formula: ROI = $\frac{\text{gross}-\text{budget}}{\text{budget}}$. This metric provides insight into the financial success of each movie. Movies with a positive ROI are considered profitable, while those with a negative ROI indicate financial losses.

4. **Distribution of ROI**: The ROI values were categorized into three groups:

   - ROI ≥ 1: 1448 movies
   - $0 \leq$ ROI < 1: 582 movies
   - ROI < 0: 656 movies

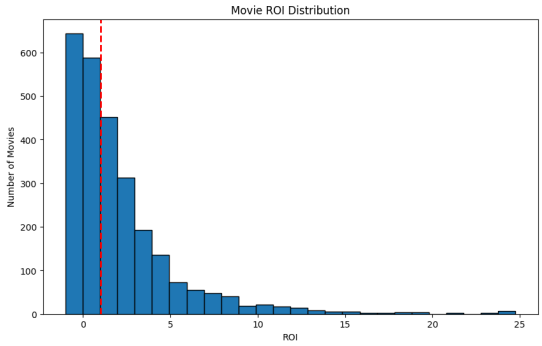

Figure 3: Distribution of ROI

3.2.2 Extracting Main Actor Information

After preprocessing the movie data, the next step involved extracting information about the main actors in each movie. This was achieved through the following steps:

1. **Splitting the `cast` field**: The `cast` field of each movie was split into a separate table to generate a list of main actors for each movie.

2. **Filtering main actors**: Only the top two main actors in each movie, represented by `order` 0 and 1, were retained for further analysis. This ensured that the dataset focused on the most prominent actors in each movie.

3.2.3 JSON File Construction

Finally, the preprocessed data was organized into a JSON file format, consisting of nodes and links. This structure facilitates the analysis of actor collaborations within the movie industry.

- **Nodes**: Each node represents an actor and includes attributes such as ID, name, gender, popularity, date of birth, and other relevant information.

- **Links**: Each link represents a collaboration between two actors in the same movie. The strength of the link is determined by the movie's ROI:

  - If ROI ≥ 1, link value = 1.
  - If $0 <$ ROI < 1, link value = 0.
  - If ROI < 0, link value = -1.

4 Methodology

To establish the recommendation system, we utilized two types of message-passing mechanism: Node-Embedding-based and GNN-based, which specifically using EGES, GCN and SEAL. The two mechanism took both graph information

and side information (e.g. *Age*) into training phase, and are expected to utilize both kinds of information to generate meaningful recommendations by link predictions.

4.1 Problem Definition

The recommendation problem can be simply viewed as a link prediction problem. While the nodes represent actors and the links are denoted as success collaboration between actors, the problem definition would be "To Predict the Probability of Success collaboration" (**Figure 4**). (Note: As mentioned in the last section, a successful collaboration is defined as ROI ≥ 1.)

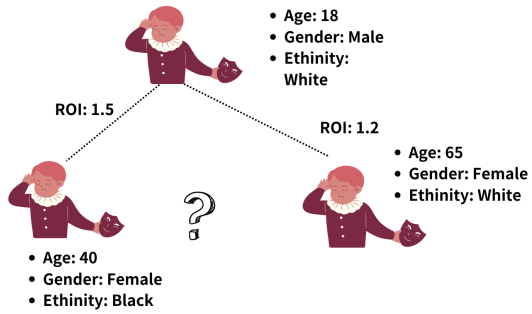

Figure 4: Illustration of Link Prediction Problem

4.2 Node-Embedding for Link Prediction

Node-Embedding-based message-passing mechanisms generates node embedding from learning node representations. In our task, the focus is to let the node embedding learning not only from graph information but also some side information of the corresponding node, while the graph remain homogeneous.

4.2.1 EGES

Enhanced Graph Embedding with Side information (EGES) is a mechanism proposed by Alibaba (Wang, 2018), and is utilized for optimizing their Taobao recommendation system. The mechanism is composed of multiple parts, including deep walk, side information weights training phase, and average pooling phase. The detailed mechanism will not be discussed here due to the focus on application. However, it is still important to mention that main concept of EGES is that different kinds (columns) of side information is not equally important to each items (which refers to the commodity on Taobao in the original paper, and refers to actor in our case), and that the items' embedding is generated by average pooling the item to every side information's embedding.

4.2.2 Implementation

Their are a few adjustments of EGES needed to be done to implement the actor network link prediction:

1. **Graph Property**: In the original paper, the graph is directed and weighted according to the order of purchase record. In the implementation, we convert our undirected graph to directed by constructing two directed link to replace one undirected link. Furthermore, all links' weight are set identical.

2. **Categorical Data**: In the original implementation, all features are categorical and one-hot encoded. Though the features in our implementation are not all categorical, we still transform those numerical features to categorical one in order to fit in the implementation by *wangzhegeek* on Github (wangzhe, 2020)

4.2.3 Link Prediction Methods

After generating embedding, we use three different link prediction methods determine whether actors should collaborate with each other.

1. **XGBoost**: Each link is composed of one source node and one target node, since it is indeed a directed graph. This method is to concatenate the source node's embedding and the target node's embedding as inputs, which are trained by XGBoost and perform link prediction.

2. **Similarity Threshold**: In this method, we calculate the cosine similarity between the source node's embedding and the target node's embedding. The similarity threshold is determined by trial and error, until the ratio of positive and negative is similar to the ground truth. This method is considered to be challenged that data leakage occurred. However, the main objective is to evaluate whether the similarity between nodes is "relatively correct in order" considering the nature of recommendation system.

3. **Similarity Ranking**: A link prediction can also be conducted by ranking similarity. This method can be adjusted according to budgets, and is easier to indicate the effectiveness of the embedding by observing the precision curve as the number of prediction increases.

4.3 GNN-based Message-Passing Mechanism

Based on node embeddings, previous research often designed diverse message passing mechanisms to effectively consider other neighboring nodes in the network. In recent years, with the mature development of Graph Neural Networks (GNNs), GNN-based methods have achieved notable results in research of recommendation systems (Zhang et al., 2018; Teru et al., 2020). Consequently, in our study, we employed two distinct message passing methods to construct our models and evaluated their performance through experiments.

4.3.1 Graph Convolutional Network with Node2Vec

Firstly, we implemented the basic Graph Convolutional Network (GCN) method. Given a adjacency matrix, G, and a set of nodes in the graph, x, we trained a set of node embeddings using Node2Vec and concatenated the features of the actors, x_{feat}, as the initial values. In this step, the node embedding utilizes the information of the entire graph, making it a transductive feature.

$$x_{emb} = E^{n2v}(G, x)$$

$$x^0_{emb} = (x_{emb} \mid\mid x_{feat})$$

Next, proceed to the convolutional layers for information passing and aggregation. In each layer, the embeddings of neighboring points from the previous layer, x^{l-1}_{emb}, are weighted and normalized using edge weights, $e_{j,i}$, and degree counts, $\hat{d}_j, \hat{d}_i$, and then aggregated to form the new representation, x^l_{emb}.

$$x_{emb}^{l} = f^{gcn}(G, x_{emb}^{l-1})$$
$$f^{gcn}(x_i) = \sum_{j \in N(i) \cup (j)} \frac{e_{j,i}}{\sqrt{\hat{d}_j \cdot \hat{d}_i}} x_j$$

After multiple iterations, a set of node embeddings, x_{emb}^{l}, is produced as the final representation. Lastly, given any node pair (i, j), the element-wise multiplication of them is then processed through a fully-connected layer, f^{fc} and a non-linear activation function, $sigmoid(\cdot)$, to obtain the probability of link existence between them, $y_{i,j}$.

$$y_{i,j} = sigmoid(f^{fc}(x_{emb\ i}^{l} \cdot x_{emb\ j}^{l}))$$

4.3.2 Advanced Model Implementation

To explore the application of more complex model architectures for learning graph information, we implemented the SEAL (Subgraphs, Embeddings, and Attributes for Link prediction) algorithm proposed by Muhan Zhang and Yixin Chen, 2018. SEAL constructs a subgraph centered around a target link and uses the feature information obtained from the subgraph for prediction. This method is based on a key assumption that all these link prediction heuristics, such as common neighbors and Katz index, can be well approximated from local subgraphs, which has been theoretically proven and validated in empirical studies.

The SEAL algorithm contains three steps: 1) enclosing subgraph extraction, 2) node information matrix construction, and 3) GNN learning.

The first step establishes subgraphs for a set of sampled positive links (observed) and a set of sampled negative links (unobserved). Here, the hyperparameter *number_of_hops* can be adjusted to control the range of local information. The second step calculates and aggregates three types of information: Node Labeling, which calculates the proximity of each node to the center in the subgraphs and assigns a label. This label information implies the influence on link existence. The node label is further concatenated with side information (attributes) and node embeddings to form the final representation. Finally, through the information propagation method (like GCN), the model predicts whether there is a link between the node pairs.

This method, compared to simple node embeddings and message passing mechanism, further leverages the partitioning of subgraphs to obtain additional features. Simultaneously, it uses local information to more effectively eliminate noise and transmit useful information, thereby training the model more efficiently.

5 Experimental Results

We conducted experiments on the collaboration network data of movie actors using four models, which include: 1) **Baseline model** (ML-based): only using features of the two actors and predicting with XGBoost Classifier, 2) **Benchmark models** (EGES, GCN): utilizing actor and network information, 3) **Best model** (SEAL): employing a more advanced model architecture for actor collaboration link prediction. We used AUC as the model evaluation metric. Through the experiments, we aim to answer three main research questions:

RQ1: How do the models perform?

RQ2: How does side information affect

model performance?

RQ3: How does our model perform in application?

5.1 Setting

In previous research, link prediction has often been considered as a self-supervised learning problem, where observed links in the network are labeled as 1 and unobserved links are labeled as 0. However, these unobserved links actually contain two meanings: either they will not occur, or they are potential connections that have not been discovered. This mixture of two implications makes it difficult for traditional classification learning methods to establish high-quality recommendation models. Even though many studies have attempted to solve this issue using ranking or contrastive learning methods (Yu et al., 2014; Chen et al., 2023), many still resort to negative random sampling to avoid this problem (Zhang and Chen, 2018).

Our research employs classification model learning, but innovatively uses "Failure Links" as negative training samples, and reasonably splits the dataset into train, validation, and test sets. As mentioned in Section 3, we categorize the links between actors based on the ROI of the movies they collaborated in, where the links are ROI $\geq$ 1 (+1), $0 \leq$ ROI $<$ 1 (0), and ROI $<$ 0 (-1). Specifically, we treat links with ROI $<$ 0 (-1) as negative training samples, assuming that actor combinations which have failed in the past are unlikely to achieve successful collaborations in the future. On the other hand, actor combinations with $0 \leq$ ROI $<$ 1 (0) are assumed to have had mediocre performance in the past and are considered to have the same potential for future collaboration as those who have never worked together.

Link type: ROI $\geq$ 1 (+1), $0 \leq$ ROI $<$ 1 (0), ROI $<$ 0 (-1), unobserved link (x)

5.2 Experiments

Based on the aforementioned model and data split settings, we further evaluate the model's performance in recommending movie actor collaborations and address the primary research questions.

5.2.1 RQ1: How do the models perform?

According to the table, SEAL outperforms all other models, achieving an AUC of 0.801 on the test dataset. This superior performance can be attributed to more effective feature extraction and message passing mechanisms. Additionally, all models that use network information perform better than the baseline model, which only uses basic actor information. This demonstrates that in this task, the cooperation relationships of actors and their neighboring networks are indeed relevant, and using graph-based methods can achieve better results. Among the benchmark models, the node embedding method (EGES) uses network information to generate representations but still fails to learn the features that influence the existence of a link between two nodes. Through the message passing mechanism, the performance of GCN is slightly improved.

5.2.2 RQ2: How does side information affect model performance?

We further discuss the impact of actors' basic information on the performance of the models. As shown in the table, in the EGES, GCN, and SEAL models, the lack of

	Train (80%)	Valid (10%)	Test (10%)
Pos	1,051 (+1)	131 (+1)	131 (+1)
Neg	651 (-1) + 400 (x)	131 (x)	131 (x)

Table 1: Data Distribution

Model	ML-based	EGES	GCN	SEAL
Valid	0.611	0.592	0.672	**0.845**
Test	0.553	0.592	0.676	**0.801**

Table 2: Model Performance (ROC AUC)

actors' basic information as model features will damage the predictive performance of actor collaboration. This result is consistent with the basic assumption of past actor selection, which considers whether there is chemistry between actors in films related to information such as appearance, age, and ethnicity.

5.2.3 How does our model perform in application?

As a recommendation model for successful actor collaborations, the model not only needs to correctly identify existing collaborative relationships but also needs to discover potential successful collaborations in order to be applicable in practice, helping film production companies select actor combinations with potential business opportunities. We use Gradio as an interface to demonstrate the effectiveness of our model in application, using three actor combinations as examples.

In **Figure 5**, Chris Evans and Robert Downey Jr. have been co-starring as the iconic characters Captain America and Iron Man, respectively, in the *Avengers* series of films since 2012. This duo has successfully formed a highly profitable and buzz-worthy actor partnership, greatly resonating with fans. For this well-known co-stardom, our model can accurately identify existing collaborative relationships.

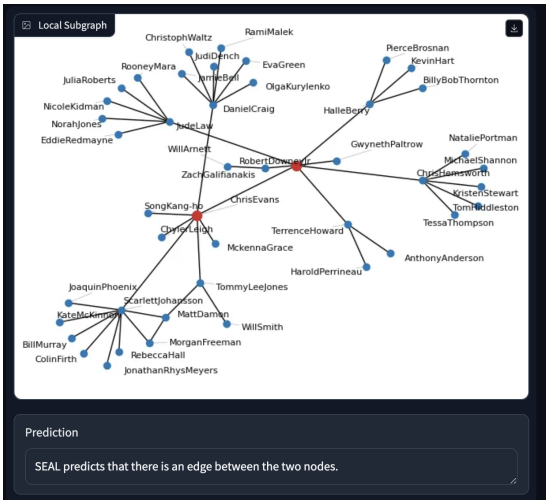

Figure 5: Prediction of Chris Evans and Robert Downey Jr

For the undiscovered potential successful collaborations, we first take Cillian Murphy and Emily Blunt as example (**Figure 6**). Cillian Murphy and Emily Blunt co-starred in the movie *A Quiet Place Part II* in 2020, which had a mediocre box office performance and was not considered a 'successful collaboration' in the dataset. However, their collaboration in *Oppenheimer*, 2023 achieved a global box office of over 9 billion US dollars and was nominated for

	Without Side Information			With Side Information		
Model	**EGES**	**GCN**	**SEAL**	**EGES**	**GCN**	**SEAL**
Valid	0.520	0.617	0.808	0.592	0.672	**0.845**
Test	0.565	0.609	0.770	0.592	0.676	**0.801**

Table 3: Performance of Models with and without Side Information

and won several film awards.

Figure 6: Prediction of Cillian Murphy and Emily Blunt

Take a look at another example (**Figure 7**), Emma Stone and Mark Ruffalo had never collaborated on a film before, but both gained significant awareness in 2023 with *Poor Things*, generating over $100 million at the box office and garnered nominations for numerous Oscars Awards.

From the above case studies, it can be seen that our model is capable of accurately identifying existing successful collaboration relationships. Moreover, for potential success of co-stardom not yet observed in the dataset, the model can recommend actor collaborations that are likely to attract audiences and generate high box office revenue based on their features and proximity in interpersonal relationships.

Figure 7: Prediction of Emma Stone and Mark Ruffalo

6 Conclusion and Future Work

In the past, when film production companies and directors were recruiting casts for a new film, they primarily considered the actors' individual popularity or their compatibility with the script. Our research offers a profit-oriented alternative, using publicly available movie and actor data to build a collaboration network and implementing a GNN-based model to achieve a 0.801 AUC score, and recommends potential successful collaborations. Nowadays, online streaming platforms, such as Netflix, Disney+, Apple TV+, have changed the business model of film industry, where the definition of a film's "success" may shift from box office earnings to viewing time on the platform,

number of times added to favorites, or discussion volume on social media.

6.1 Possible Improvements for EGES

There are a few factors that might be able to conclude the poor performance of EGES link prediction:

1. **Numeric Features are One-hot Encoded to Categorical Features**: Some of the numeric features, which could be representative, are one-hot encoded to categorical features. By doing so, the continuous and ordinal nature of numeric features are not learned by the model. Improvements can be made by modifying the model or group numeric features into intervals, which may have better performance comparing to one-hot encoding.
2. **Unweighted Graph**: In the original paper, the graph is weighted according to the transaction record. In our case, popularity could also serve as weight, which might help the model to distinguish important features and links.

6.2 Possible Improvements for the Overall Method

In future work, if we could obtain internal data from online video streaming platforms, we could extend our method by adjusting the definition of successful links to better align with the platform's business model. Additionally, a movie's success is not solely due to the cast; it may also be attributed to factors such as the story, director, special effects, and others. By analyzing online movie reviews, we can determine whether the cast is a major factor in the success of a movie, consolidating the association between a movie's success and the involvement of its main actors. Many actors focus their careers on similar categories of movie. By considering movie genres, we can establish more refined linkages and further study the benefits of diversified, cross-genre collaborations among actors. Finally, in addition to the practical optimization mentioned above, consideration of temporal relation and heterogeneous information may also enhance the performance of recommendation models.

References

[1] Nelson, R. A., Glotfelty, R. (2012). Movie stars and box office revenues: an empirical analysis. *Journal of Cultural Economics, 36, 141-166.*

[2] Kwan, A. W., Scheepers, S. (2022). The Fault in Our Stars: A Quantitative Study on the Effect of Cast Member Celebrity on Film Success. *Journal of Student Research, 11(2).*

[3] Zhu, Z., Zhang, Z., Xhonneux, L. P., Tang, J. (2021). Neural bellman-ford networks: A general graph neural network framework for link prediction. *Advances in Neural Information Processing Systems, 34, 29476-29490.*

[4] Chamberlain, B. P., Shirobokov, S., Rossi, E., Frasca, F., Markovich, T., Hammerla, N. Y., ... Hansmire, M. (2022). Graph neural networks for link prediction with subgraph sketching. In *The eleventh international conference on learning representations.*

[5] Perozzi, B., Al-Rfou, R., Skiena, S. (2014). Deepwalk: Online learning of so-

ID	Member	Work
R11725041	Jih-Ming Bai	Literature Research, Models and Experiment
R12725028	Cheng-Yu Kuan	Literature Research, Demo
B10704096	Shang-Qing Su	Literature Research, Data Collection
B10704031	Po-Yen Chu	Literature Research, EGES Model Building, Experiment
B10704051	Chia-Shan Li	Literature Research, Data Collection Slides

Table 4: Work Division

cial representations. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining (pp. 701-710).*

[6] Tang, J., Qu, M., Wang, M., Zhang, M., Yan, J., Mei, Q. (2015). Line: Large-scale information network embedding. In *Proceedings of the 24th international conference on world wide web (pp. 1067-1077).*

[7] Grover, A., Leskovec, J. (2016). node2vec: Scalable feature learning for networks. In *Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining (pp. 855-864).*

[8] Ribeiro, L. F., Saverese, P. H., Figueiredo, D. R. (2017). struc2vec: Learning node representations from structural identity. In *Proceedings of the 23rd ACM SIGKDD international conference on knowledge discovery and data mining (pp. 385-394).*

[9] Dong, Y., Chawla, N. V., Swami, A. (2017). metapath2vec: Scalable representation learning for heterogeneous networks. In *Proceedings of the 23rd ACM SIGKDD international conference on knowledge discovery and data mining (pp. 135-144).*

[10] Zhang, M., Chen, Y. (2018). Link prediction based on graph neural networks. *Advances in neural information processing systems, 31.*

[11] Teru, K., Denis, E., Hamilton, W. (2020). Inductive relation prediction by subgraph reasoning. In *International Conference on Machine Learning (pp. 9448-9457). PMLR.*

[12] McAuley, J., University of California, San Diego, *Recommender Systems and Personalization Datasets*. Accessed Dec 23, 2023, `https://cseweb.ucsd.edu/~jmcauley/datasets.html#social_data`.

[13] Cai, C., He, R., McAuley, J. (2017). SPMC: Socially-Aware Personalized Markov Chains for Sparse Sequential Recommendation. *arXiv preprint arXiv:1708.04497.*

[14] Zhao, T., McAuley, J., King, I. (2015). Improving Latent Factor Models via Personalized Feature Projection for One-Class Recommendation. In *Proceedings of the ACM International Conference on Information and Knowledge Management (CIKM), pp. 821–830.*

[15] Wang, J., Huang, P., Zhao, H., Zhang, Z., Zhao, B., Lee, D. L. (2018). Billion-scale Commodity Embedding for E-

commerce Recommendation in Alibaba. *arXiv:1803.02349*.

[16] wangzhe. (2020) *EGES*. `https://github.com/wangzhegeek/EGES`.