

## Article

# Comparison of Graph Distance Measures for Movie Similarity Using a Multilayer Network Model

Majda Lafhel <sup>1,\*</sup>, Hocine Cherifi <sup>2,†</sup>, Benjamin Renoust <sup>3,†</sup> and Mohammed El Hassouni <sup>4,†</sup><sup>1</sup> FLSH, LRIT, FS, Mohammed V University in Rabat, Rabat 10090, Morocco<sup>2</sup> ICB UMR 6303 CNRS, University of Burgundy, 21000 Dijon, France; hocine.cherifi@u-bourgogne.fr<sup>3</sup> Institute for Dataability Science, Osaka University, Osaka 565-0871, Japan; renoust@ids.osaka-u.ac.jp<sup>4</sup> FLSH, Mohammed V University in Rabat, Rabat 10090, Morocco; mohamed.elhassouni@flsh.um5.ac.ma

\* Correspondence: majdalafhel1@gmail.com

† These authors contributed equally to this work.

**Abstract:** Graph distance measures have emerged as an effective tool for evaluating the similarity or dissimilarity between graphs. Recently, there has been a growing trend in the application of movie networks to analyze and understand movie stories. Previous studies focused on computing the distance between individual characters in narratives and identifying the most important ones. Unlike previous techniques, which often relied on representing movie stories through single-layer networks based on characters or keywords, a new multilayer network model was developed to allow a more comprehensive representation of movie stories, including character, keyword, and location aspects. To assess the similarities among movie stories, we propose a methodology that utilizes a multilayer network model and layer-to-layer distance measures. We aim to quantify the similarity between movie networks by verifying two aspects: (i) regarding many components of the movie story and (ii) quantifying the distance between their corresponding movie networks. We tend to explore how five graph distance measures reveal the similarity between movie stories in two aspects: (i) finding the order of similarity among movies within the same genre, and (ii) classifying movie stories based on genre. We select movies from various genres: sci-fi, horror, romance, and comedy. We extract movie stories from movie scripts regarding character, keyword, and location entities to perform this. Then, we compute the distance between movie networks using different methods, such as the network portrait divergence, the network Laplacian spectra descriptor (NetLSD), the network embedding as matrix factorization (NetMF), the Laplacian spectra, and D-measure. The study shows the effectiveness of different methods for identifying similarities among various genres and classifying movies across different genres. The results suggest that the efficiency of an approach on a specific network type depends on its capacity to capture the inherent network structure of that type. We propose incorporating the approach into movie recommendation systems.

**Keywords:** movie; multilayer network; network similarity; movie genre classification; network quantification; graph distance measure



**Citation:** Lafhel, M.; Cherifi, H.; Renoust, B.; El Hassouni, M. Comparison of Graph Distance Measures for Movie Similarity Using a Multilayer Network Model. *Entropy* **2024**, *26*, 149. <https://doi.org/10.3390/e26020149>

Academic Editors: Adam Lipowski and Alessandro Pluchino

Received: 4 December 2023

Revised: 1 February 2024

Accepted: 3 February 2024

Published: 8 February 2024



**Copyright:** © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

## 1. Introduction

A recommendation system is a filtering system that suggests a list of items similar to a user's favorites. Nowadays, recommendation systems are integrated everywhere on many platforms, such as e-commerce, social media, YouTube, etc. A movie recommendation system, such as Netflix, proposes to users a set of movies based on filtering their data or recent online activities. Recommendation systems employ filtering techniques [1–4] such as collaborative-based, content-based, and hybrid-based methods. However, these techniques rely on algorithms for collecting and analyzing user information or interactions through web navigation, which leads to security and privacy issues. An alternative way to identify similar movies without accessing user data is by assessing movie stories represented as networks.

Over the last few years, the analysis of movies has increasingly become a crucial aspect and a challenging issue in complex network analysis. The process involves identifying different components of a movie story, converting them into networks, and then selecting appropriate approaches to measure and evaluate these networks. Studies were conducted to bridge the semantic gap through summarization [5,6] and audiovisual information [7]. Adams et al. (2002) [8] used social networks to sort movies into categories. Weng et al. (2009) [9] and Jung et al. (2013) [10] investigated movie stories using social networks based on character interactions. All previous studies have been conducted to analyze characters. The most widespread method is representing characters in a single or a bipartite graph [11]. However, one component of the story is not sufficiently efficient to give a comprehensive overview of the story. For this, Mouchid et al. [12] proposed a multilayer network model to capture more elements of the movie story, including characters, keywords, and locations.

Markovic et al. (2019) [13] conducted a study to construct the character network of Slovene belles-lettres based on its interaction structures. To evaluate their interactions, they created a list of main characters and indexed sentences in which they appear. The distance between characters is then computed based on their frequency of occurrence in the text. Lv et al. (2018) [14] proposed StoryRoleNet, an algorithm to construct the character network of a movie from its corresponding video and subtitle. Then, they identified the main characters using the Louvain algorithm for community detection. In another study, Chen et al. (2019) [15] suggested using the minimum span clustering algorithm on community structures and centrality to find the distance between characters extracted from the novel *Dream of the Red Chamber*. All these studies aim to calculate the distance between individual characters in a narrative and identify the most important ones. Mouchid et al. (2019) [16] proposed visualizing characters, keywords, locations, captions, and faces in *Star Wars* using community detection. They identified the top ten nodes within individual layers by comparing influence scores of their corresponding diameter, number of nodes, number of edges, clustering coefficient, shortest path, assortativity, number of communities, and modularity. However, despite considering many components of the movie story, they also rely on analyzing individual characters by highlighting the main ones. To the best of our knowledge, there is currently no approach for measuring the similarity between movie stories, verifying two aspects: (i) regarding many components of the movie story and (ii) quantifying the distance between corresponding movie networks. Thus, we aim to quantify the similarity between a couple of movies by assessing the distance of their corresponding networks considering many elements. We rely on the multilayer network model [12] to extract three-layer entities, i.e., character, keyword, and location. We then compute the distance between these layers using graph distance techniques. Determining the distance between networks is a challenging task in network science, as a distance measure may work well on one type of graph but not on another, depending on the network structure and topology. Our primary objective is to investigate the effectiveness of graph distance measures for estimating movie similarity so that they can be incorporated into recommendation systems.

Graph distance techniques involve two main steps: (i) extracting network feature vectors and (ii) computing the distance between them. In the context of network analysis, many authors rely on techniques such as node degree visualization [17], centrality visualization [18], and community visualization [19]. In general, there are two techniques for analyzing a network [20]. The first method concerns presenting network data using network visualization such as diagrams, heat maps, or graph displays. The visual comparison allows us to make a general overview and assumptions about the network. Numerous visual tools allow for exploring networks, such as Gephi [21]. The second technique involves extracting network properties, such as node degrees, graphlets, or centrality. These properties help in investigating the structure of a network. In our study, we opt to focus on the second category due to its precision.

Schieber [22] combined three features of probability distribution functions—node degree, node dispersion, and node alpha-centrality—into a single vector. He first calculated

the distribution of each of the three features and then used Jensen–Shannon to compute the distance between these probability distributions. Ronda [23] (2020) extracted node connectivity and node similarity features and computed the distance between vectors using cosine similarity. Saxena [24] (2019) extracted k-core, k-truss, and node degrees to analyze the hierarchy level and the assortativity. The GRAAL (GRAPh ALigner) family (M-GRAAL, L-GRAAL, C-GRAAL, and H-GRAAL) is used for biological network alignment, except MI-GRAAL, which can analyze topological features. Brodka [25] (2018) classified distance measures into three categories: (i) transform property vectors into scalar values and measure their relative differences; (ii) compute the frequency distributions of the property vectors and determine the distance between them; (iii) compare the property vector using a measure of overlapping or correlation.

Two major categories of methods can be distinguished: known node-correspondence and unknown node-correspondence. The first category involves comparing networks that require prior knowledge of the nodes, such as graphs with the same size, node labels, node-set, and edge-set. In contrast, the second category compares networks that do not necessitate prior knowledge of nodes, providing valuable insights into the structure and topology of graphs. Our research focuses on node-correspondence methods as we work on networks of different sizes.

In our previous work [26], we analyzed the similarity in the 6-cycle Star War Saga (SW) movies. We used the multilayer network model to extract character, keyword, and location networks. Then, we employed network portrait divergence [27] to compute the distance between movie layers. Our findings suggest a high similarity among characters in both the prequel and sequel trilogies and a notable distinction in locations between both trilogies. Moreover, there is a significant similarity in locations within each of the prequel and sequel trilogies. The results reveal similarities in the relationships between topics (keywords) in Star Wars episodes II (SW2) and III (SW3), as well as in episode I (SW1) with episodes IV (SW4) and VI (SW6). However, other episodes exhibit dissimilarities, particularly the relationship connecting keywords of episode V (SW5) with episodes II (SW2) and III (SW3). In recent work [28], we studied the efficiency of NetLSD (network Laplacian spectral descriptor) in revealing the similarity between the 3-cycle movies of the Scream Saga (SC). The analysis indicates higher similarity between keywords in episodes I and II than in episode III. Moreover, the findings validate a degradation in the similarity among the characters and locations across the episodes. In the current work, we investigate the performance of more distance measures, namely, the network matrix factorization, the Laplacian spectra, and the D-measure for comparing movie networks from four categories: sci-fi, horror, romance, and comedy. Moreover, we investigate the performance of distance measures in categorizing movie genres.

The rest of this paper is organized as follows. In Section 2, we summarize the multilayer movie script model and its extraction process. Section 3 describes the approaches used for comparing movie networks. Section 4 describes the *dataset* and the ground truth data. In Section 6, we apply the measures to the movie networks, interpreting the results. We conclude in Section 7.

## 2. Multilayer Movie Model

Mourchid et al. [12] proposed a multilayer network model to capture the different elements of a movie story and answer the most commonly asked questions in film narration—Who?, Where?, and What? According to their model, the characters answer the question Who, the keywords answer the question What, and the locations answer the question Where. The multilayer network consists of three layers, each representing a different entity and interaction. In the following, we describe the concept of the constitution of the network model.

## 2.1. Definition

### Nodes

Each layer contains nodes belonging to the same category. There are three sets of nodes (character, location, keyword), which are defined as follows:

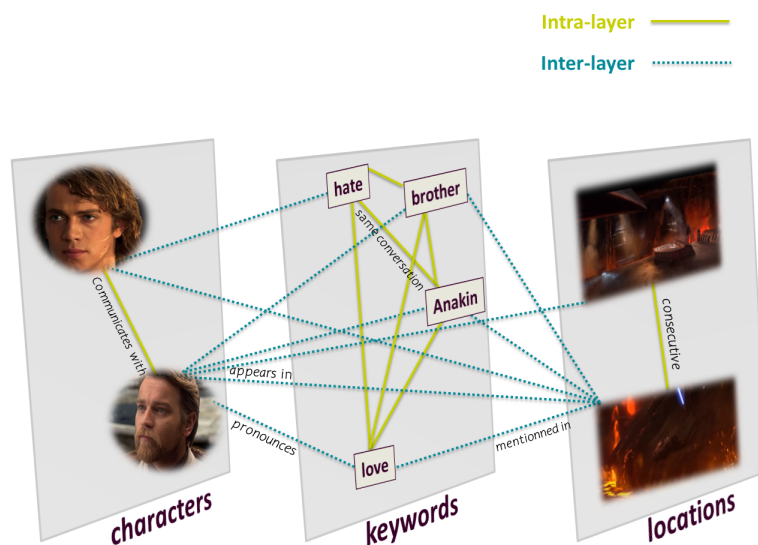
- A character refers to an actor in a movie.
- A location is a place where a scene turns.
- A keyword is a significant word uttered by a character during dialogues.

### Links

There are two types of links (intralayer and interlayer), which are defined as follows:

- **Intralayer link** connects nodes of the same entity:
  - A link when two characters communicate with each other.
  - A link when two locations are consecutive.
  - A link when two keywords belong to the same conversation.
- **Interlayer link** connects nodes of different entities:
  - A link between a character and a location if the character appears in the location in a scene.
  - A link between a character and a keyword if the character pronounces the keyword.
  - A link between a location and a keyword if the keyword is mentioned in a conversation taking place in the location.

The multilayer network includes numerous nodes and relationships. However, Figure 1 shows a sample of the multilayer network model for a better understanding of interactions.



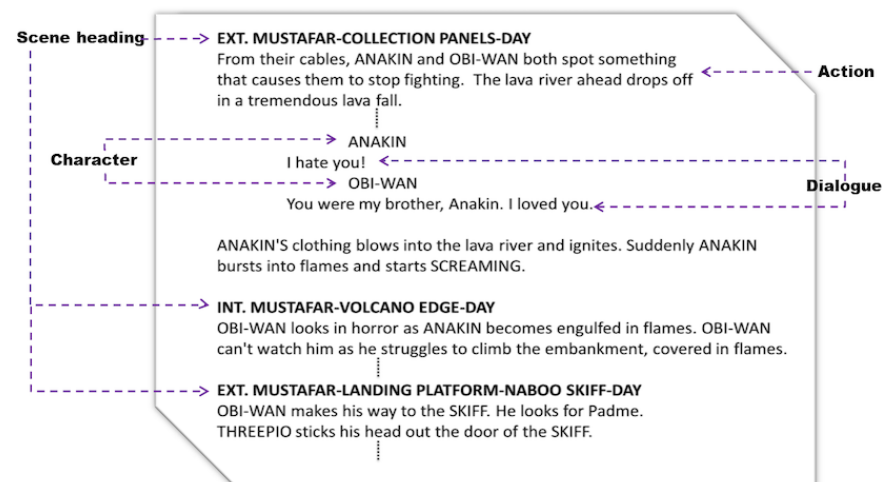
**Figure 1.** Multilayer network model for extracting movie stories. The green line represents intralayer links connecting nodes of the same entity. The blue dotted line represents interlayer links connecting nodes of the same entity.

## 2.2. Network Extraction

We explain in this section how to extract the multilayer network model from a movie script by identifying entities and their interactions. Firstly, we append a glossary of the semantic components found in the movie script.

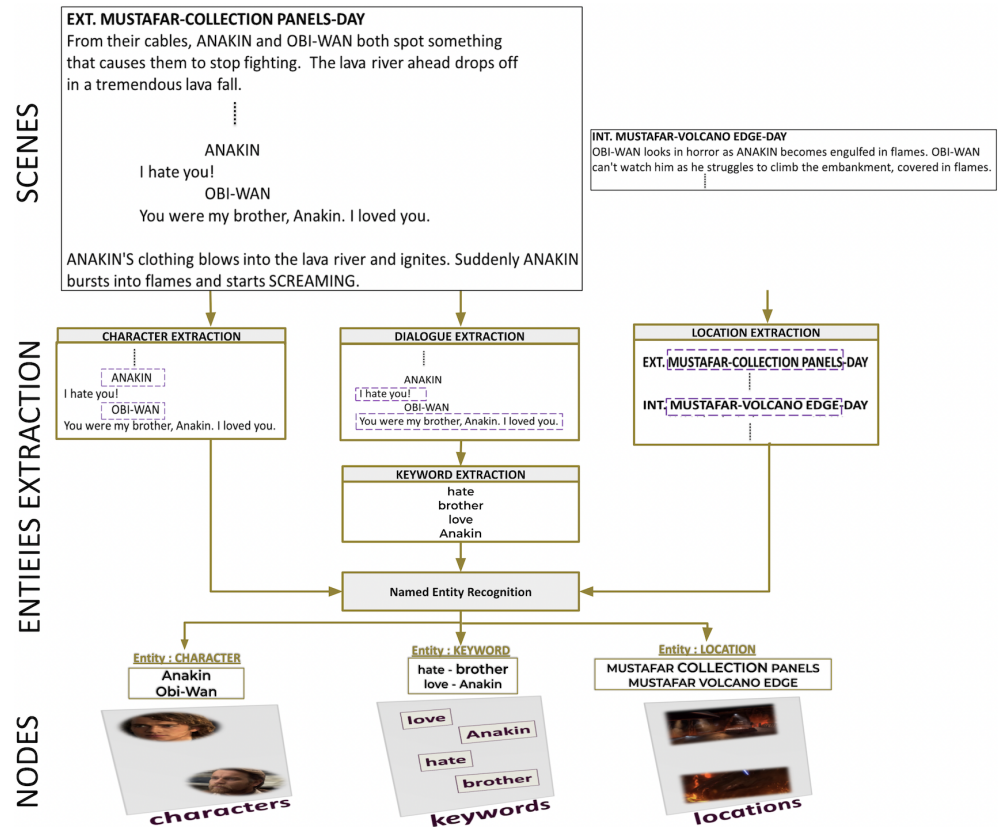
Term	Definition
Script (Screenplay)	A document includes technical information about scenes, dialogues, and settings.
Scene heading	The start of a scene in a screenplay. A scene heading describes the physical spaces (INT or EXT), location, and time of the day (DAY or NIGHT).
Scene	A piece of the script. Each script is divided into scenes, which are separated by scene headings.
Dialogue	The lines of a speech a character must say in a scene.
Conversation	An interchange of dialogue between two or more characters in a script.
Action	Lines describe visual and audible actions in a scene.

Figure 2 displays a piece from the movie script *Revenge of the Sith*. Within this snippet, three scenes are delineated by scene headings, with the latter consistently followed by action lines. Character names are written in uppercase and positioned before dialogue lines. Note that some scenes contain only action lines.

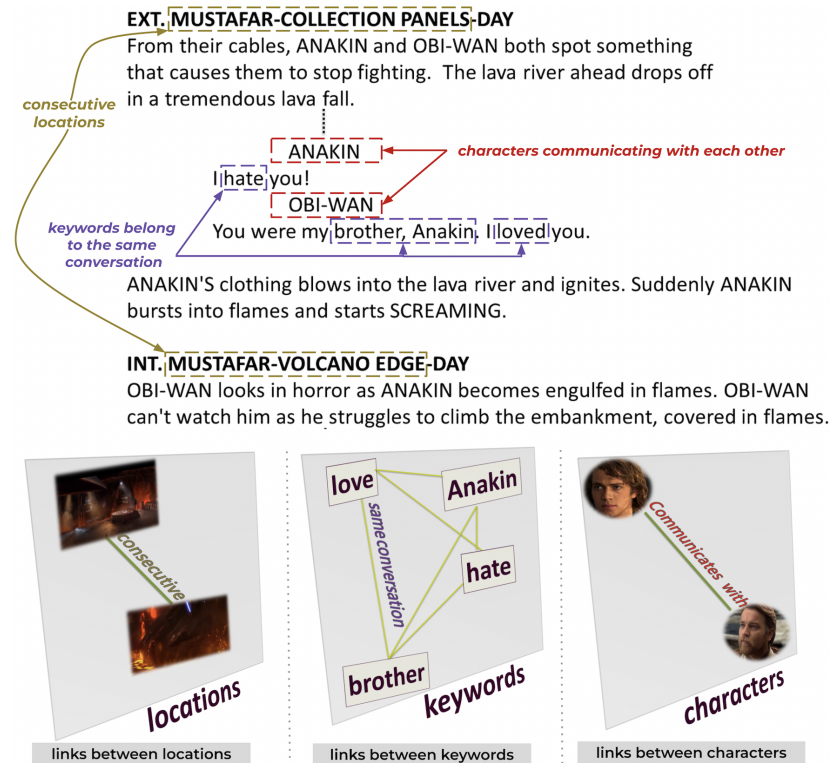


**Figure 2.** A piece of the script extracted from the movie ‘Attack of the Clones’. The figure illustrates elements of a movie script: Scene Heading, Character, Action, and Dialogue.

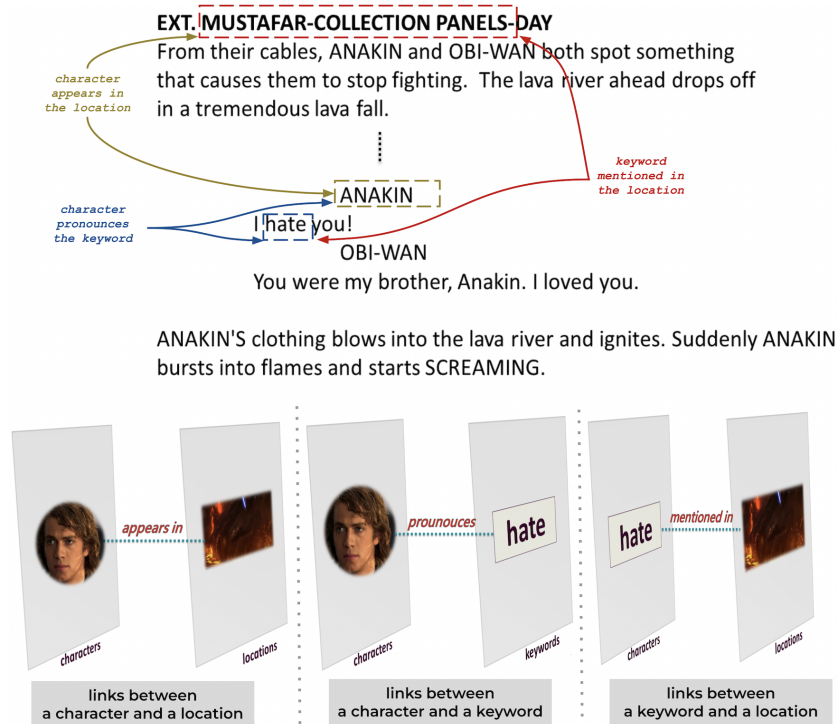
The first step of the process is to chunk the script into scenes. Figure 3 provides an overview of extracting entities from the script. Due to the straightforward structure of the text, identifying locations and characters is a simple task during analysis. For instance, consider the first scene. The location *MOSTAFAR COLLECTION PANELS* in the scene heading is placed just after the physical space *EXT.* *ANAKIN* and *OBI-WAN* are the characters. The keywords are retrieved from dialogues using the latent Dirichlet allocation (LDA) [29] method. Named entity recognition (NER) is a tool used to identify different types of entities: characters, keywords, and locations within action lines. Figures 4 and 5 provide an example of the extraction of intralayer and interlayer links, respectively.



**Figure 3.** The process of extracting the entities: characters, keywords, and locations. First, the script is divided into scenes. Locations are extracted from scene headings, keywords from dialogues, and characters from the lines preceding dialogues. Then, named entity recognition is applied to classify them into entities.



**Figure 4.** Extraction of intralayer links: An intralayer link is established between characters who communicate in the same scene, keywords that belong to the same conversation, or consecutive locations.



**Figure 5.** Extraction of interlayer links: Interlayer links are established between a character and a location if the character appears in the location, between a character and a keyword if the character mentions the keyword, or between a location and a keyword if the keyword is discussed in a conversation within the location.

### 3. Network Similarity Measures

Unknown-node correspondence can be categorized into three main approaches: spectral, embedding, and statistical. In this section, we present the measures used in experimentation for each category. The Table 1 illustrates the terms and notations used in this paper.

**Table 1.** Terms and Notations.

Symbol	Description
$G$	Undirected and unweighted network
$V, n$	Set of vertices, Number of nodes
$E, m$	Set of edges, Number of edges
$A$	$n \times n$ adjacency matrix
$D$	$n \times n$ diagonal matrix
$I$	$n \times n$ identity matrix
$L$	Laplacian matrix
$\tilde{L}$	Normalized Laplacian matrix
$\rho$	Orthogonal matrix
$\zeta$	Graph representation
$d$	Graph's diameter
$\lambda$	Eigenvalue
$v$	Feature vector
$\tilde{D}$	Distance
$JS$	Jensen–Shannon divergence

#### 3.1. Spectral Methods

Two graphs are supposed to be isomorphic if they are isospectral; in other words, if they share the same spectrum [30]. However, this hypothesis is doubtful because two dif-

ferent networks can have the same spectrum [31]. Nevertheless, numerous investigations are underway to solve this problem. Given a graph  $G = (V, E)$ , where  $V$  is a set of vertices and  $E \subseteq V \times V$  is a set of edges,  $G$  can be represented as a square matrix  $M$  of size  $n \times n$ , where  $M$  encodes the structural properties of a graph, such as the node degree.  $M$  can be the adjacent matrix, Laplacian matrix, normalized Laplacian matrix, or heat kernel matrix. The spectrum of a matrix  $M$ , denoted  $s$ , is the set of eigenvalues  $\{\lambda_1, \lambda_2, \dots, \lambda_{|V|}\}$ , where an eigenvalue  $\lambda_i$  is the root of the characteristic polynomial  $\mathcal{P}_M$  associated with  $M$ . Eigenvalues are obtained by solving the polynomial equation  $\mathcal{P}_M(\lambda) = \det(\lambda I - M) = 0$ , where  $I$  represents the  $n \times n$  identity matrix.

The matrix representation  $M$  can be written as the eigendecomposition  $M = \rho D \rho^{-1}$ , where

$$D = \begin{pmatrix} \lambda_1 & 0 & \cdots & 0 \\ 0 & \lambda_2 & \cdots & 0 \\ \vdots & & & \\ 0 & 0 & \cdots & \lambda_{|V|} \end{pmatrix}$$

is the diagonal matrix whose entry is the eigenvalues and  $\rho = [\rho_1, \rho_2, \dots, \rho_{|V|}]$  is the orthogonal matrix whose a column  $\rho_i$  is the corresponding eigenvector of the eigenvalue  $\lambda_i$ . Thus, it is possible to derive a spectrum from either an eigendecomposition of  $M$  or the characteristic polynomials  $\mathcal{P}_M$ .

### 3.1.1. Euclidean Distance between Spectra

Wilson and Zhu [30] have demonstrated that the Laplacian matrix is more effective than the adjacency and normalized Laplacian matrices in clustering and classification. Furthermore, the Laplacian spectra show a lower occurrence of the isospectrality issue than the adjacency spectra or normalized Laplacian spectra [30]. Laplacian spectra is a permutation-invariant and scale-adaptive measure. The distance between the spectra precises if networks are similar or dissimilar. Let  $v_{s_{G_A}} = (\lambda_{A_1}, \lambda_{A_2}, \dots, \lambda_{A_n})$  and  $v_{s_{G_B}} = (\lambda_{B_1}, \lambda_{B_2}, \dots, \lambda_{B_n})$  be vectors, including the spectra of the graphs  $G_A$  and  $G_B$ , respectively. Euclidean distance between  $s_{G_A}$  and  $s_{G_B}$  is determined using the difference between  $G_A$  and  $G_B$ . It is defined as follows:

$$\tilde{\mathcal{D}}(s_{G_A}, s_{G_B}) = \sqrt{(\lambda_{A_1} - \lambda_{B_1})^2 + (\lambda_{A_2} - \lambda_{B_2})^2 + \dots + (\lambda_{A_n} - \lambda_{B_n})^2} = \sqrt{\sum_{i=0}^n (\lambda_{A_i} - \lambda_{B_i})^2} \tag{1}$$

Algorithm 1 shows the steps for computing the distance  $\tilde{\mathcal{D}}$  between two networks across Laplacian spectra.

---

**Algorithm 1** Compute the distance between two networks across Laplacian spectra.

---

input:  $G_A$  and  $G_B$

output: single value

    Compute  $L_A = \text{Laplacian\_matrix}(G_A)$  //Return the Laplacian Matrix of  $G_A$

    Compute  $L_B = \text{Laplacian\_matrix}(G_B)$  //Return the Laplacian Matrix of  $G_B$

    Compute  $v_{s_{G_A}} = \text{spectra}(L_A)$  //Extract Spectra of  $L_A$  as a vector

    Compute  $v_{s_{G_B}} = \text{spectra}(L_B)$  //Extract Spectra of  $L_B$  as a vector

    return  $\tilde{\mathcal{D}}(v_{s_{G_A}}, v_{s_{G_B}})$  //The output is a single value

---

### 3.1.2. Network Laplacian Spectral Descriptor

The network Laplacian spectral descriptor (NetLSD) [32] is a recent method for graph comparison. Given a graph  $G$  with  $n$  nodes, NetLSD derives the  $n$ -dimensional vector  $u_t$  from the heat equation  $\frac{\partial u_t}{\partial t} = -\tilde{L}u_t$  where  $u_t$  is the heat properties of nodes. The closed-form solution is  $n \times n$  heat kernel matrix  $h_t$ , such as



$$h_t = e^{-t\tilde{L}} \tag{2}$$

The heat matrix  $h_t$  verifies three properties: permutation-invariant, scale-adaptive, and size-invariant. In the following, we will elaborate on these three properties and illustrate how  $h_t$  accomplishes them.

- **Permutation-invariant:** A distance  $\tilde{D}$  on a graph representation  $\zeta$  is permutation-invariant if, despite permuting two given networks  $G_A$  and  $G_B$ , their graph representations  $\zeta$  remain identical:

$$\forall G_A, G_B \quad G_A \simeq G_B \Rightarrow \tilde{D}(\zeta(G_A), \zeta(G_B)) = 0 \tag{3}$$

As seen in Equation (2), we note that the  $h_t$  inherits properties from  $\tilde{L}$ . Since the normalized Laplacian spectrum ( $\tilde{L}$ ) verifies the permutation property, the heat matrix( $h_t$ ) also verifies the permutation property.

- **Scale-adaptive:** A graph representation  $\zeta$  is scale adaptive if it contains both local feature  $\varphi_l$  and global feature  $\varphi_g$ :
  - Local feature ( $\varphi_l$ ) captures information about the graph structure at the local level:

$$\forall G, \exists f(\cdot), \quad \varphi_l = f(\zeta(G)) \tag{4}$$

- Global feature ( $\varphi_g$ ) captures information about the graph structure at the global level:

$$\forall G, \exists f(\cdot), \quad \varphi_g = f(\zeta(G)) \tag{5}$$

The heat kernel matrix can encode global and local connectivities thanks to its diagonal matrix.

- **Size-invariant:** Let  $\Delta$  be a domain. A distance  $\tilde{D}$  on a graph representation  $\zeta$  is size-invariant if it verifies:

$$\forall \Delta : G_A, G_B \text{ sampled from } \Delta \quad \Rightarrow \quad \tilde{D}(\zeta(G_A), \zeta(G_B)) = 0 \tag{6}$$

Regardless of the shape of  $G_A$  and  $G_B$ , if they are sampled from the same domain  $\Delta$ , the distance  $\tilde{D}$  between their graph representations should be equal to 0. The heat kernel matrix fulfills the size-invariant property, as demonstrated by the authors [32], who proved the ability of the heat kernel to output comparable values even for complete and empty graphs.

After extracting heat kernel matrices  $h_{t_A}$  and  $h_{t_B}$  for networks  $G_A$  and  $G_B$ , they are reshaped into vectors  $v_{h_{t_A}}$  and  $v_{h_{t_B}}$ , respectively. Then, the Euclidean distance (Equation (1)) is used to compute the distance between heat kernel vectors  $v_{h_{t_A}}$  and  $v_{h_{t_B}}$ . Algorithm 2 shows the steps for computing the distance  $\tilde{D}$  between two networks across NetLSD.

---

**Algorithm 2** Compute the distance between two networks across NetLSD.

---

input:  $G_A$  and  $G_B$

output: single value

```

Compute  $h_{t_A} = \text{NetLSD}(G_A)$  //Return the NetLSD of  $G_A$  as a matrix
Compute  $h_{t_B} = \text{NetLSD}(G_B)$  //Return the NetLSD of  $G_B$  as a matrix
Compute  $v_{h_{t_A}} = \text{Reshape}(h_{t_A})$  //Convert heat kernel matrix  $h_{t_A}$  into vector
Compute  $v_{h_{t_B}} = \text{Reshape}(h_{t_B})$  //Convert heat kernel matrix  $h_{t_B}$  into vector
return  $\tilde{D}(v_{h_{t_A}}, v_{h_{t_B}})$  //The output is a single value
    
```

---

### 3.2. Embedding Methods

In the literature [33–35], the term embedding has been used in two ways: graph embedding or node embedding. Graph embedding is a technique that involves mapping the nodes of a network into a low-dimensional vector, while node embedding involves mapping each node to a particular vector. During the past decade, network embedding has been extensively used in node classification [36,37], clustering [38–40], community detection [41], visualization [42], and network comparison [43–45]. Two nodes are regarded to be similar if they are positioned closer to each other in space. The main important feature in graph embedding is the *order-proximity*. An efficient network embedding method should verify both the first-order proximity, which is determined by the edge weight between two nodes  $v_i$  and  $v_j$ , and the second-order proximity, which is determined by the similarity between the neighbors of nodes  $v_i$  and  $v_j$ . The prominent challenge of graph embedding techniques is to preserve the network structure [33,34].

A plethora of graph-embedding techniques is available [33]. The matrix factorization technique is useful in recommendation systems [46]. It is efficient and has important features. The matrix factorization is related to the singular value decomposition (SVD) [47] technique which decomposes a matrix  $\mathcal{M}$  into three matrices  $\mathcal{M} = \rho D \rho^T$ , where  $\rho$  is the orthogonal matrix of  $\mathcal{M}$  and  $D$  is its diagonal matrix. The SVD provides a unique solution  $D$  for the equation  $\mathcal{M} = \rho D \rho^T$  as it extracts unique feature singular values.

NetMF (network embedding as matrix factorization) is a recent permutation-invariant network representation learning model used for graph embedding. NetMF employs the network matrix-factorization-based technique [46] for embedding DeepWalk [48]. Jiezhong et al. [49] concluded the closed-form of DeepWalk as matrix factorization  $(\frac{1}{W} \sum_{r=1}^W P^r) D^{-1}$ . They then demonstrated a relationship between the closed form of the DeepWalk and the normalized Laplacian matrix  $\tilde{L}$ , such as  $D^{-1/2} A D^{-1/2} = I - \tilde{L}$ .

The NetMF algorithm takes as input a network  $G$  and produces as output a matrix  $n \times n$  representing the network embedding  $Q$ . To compare two networks  $G_A$  and  $G_B$ , we first extract their corresponding matrices  $Q_A$  and  $Q_B$ . Second, we reshape  $Q_A$  and  $Q_B$  into vectors  $v_{Q_A}$  and  $v_{Q_B}$ . Then, the Euclidean distance (Equation (1)) is used to compute the distance between embedding vectors  $v_{Q_A}$  and  $v_{Q_B}$ . Algorithm 3 shows the steps for computing the distance  $\tilde{D}$  between two networks across their network embeddings.

---

**Algorithm 3** Compute the distance between two networks across network NetMF.

---

input:  $G_A$  and  $G_B$

output: single value

```

Compute  $Q_A = \text{NetMF}(G_A)$  //Return the Network Embedding of  $G_A$  as a matrix
Compute  $Q_B = \text{NetMF}(G_B)$  //Return the Network Embedding of  $G_B$  as a matrix
Compute  $v_{Q_A} = \text{Reshape}(Q_A)$  //Convert Network Embedding  $Q_A$  into vector
Compute  $v_{Q_B} = \text{Reshape}(Q_B)$  //Convert Network Embedding  $Q_B$  into vector
return  $\tilde{D}(v_{Q_A}, v_{Q_B})$  //The output is a single value

```

---

### 3.3. Statistical Methods

A statistical method describes a network by probing its characteristic properties. The primary step in statistical approaches is extracting network features, such as node degrees, degree distribution, shortest path, etc. Features can be represented as singular values, vectors, or matrices. The second step consists of computing the distance between them.

#### 3.3.1. Portrait Divergence

Network portrait divergence [50] is a permutation-invariant measure used to compare two complex networks based on the probability distribution feature and the Jensen–Shannon divergence. The network portrait [27] is a matrix  $\mathcal{B}$  where each row represents the probability distribution  $P(k|l)$ , such as:

$$P(k|l) = \frac{1}{N} \mathcal{B}_{l,k} \tag{7}$$

where  $k$  is the number of nodes accessible at distance  $l$  from a randomly chosen node.

In two steps, the network portrait divergence computes the distance between two networks  $G_A$  and  $G_B$ . First, it calculates the probability distributions  $P_{\mathcal{B}_A}$  and  $P_{\mathcal{B}_B}$  of  $G_A$  and  $G_B$ , relying on Equation (7). At the end of this step,  $G_A$  and  $G_B$  are associated with the network portraits  $\mathcal{B}_A$  and  $\mathcal{B}_B$ , respectively. Second, network portrait divergence computes the distance between the network portraits  $\mathcal{B}_A$  and  $\mathcal{B}_B$  using the Jensen–Shannon divergence, such as

$$\tilde{\mathcal{D}}_{JS}(G_A, G_B) = \frac{1}{2}(KL(P_{\mathcal{B}_A}||P_*) + KL(P_{\mathcal{B}_B}||P_*)) \tag{8}$$

where  $P_* = \frac{(P_{\mathcal{B}_A} + P_{\mathcal{B}_B})}{2}$ , and  $KL(\cdot||\cdot)$  is the Kullback–Liebler divergence between two probability distributions  $P_{\mathcal{B}_A}$  and  $P_{\mathcal{B}_B}$ , such as

$$KL(P_{\mathcal{B}_A}(k|l)||P_{\mathcal{B}_B}(k|l)) = \sum_{l=0}^{\max(d_A, d_B)} \sum_{k=0}^N P_{\mathcal{B}_A}(k|l) \log\left(\frac{P_{\mathcal{B}_A}(k|l)}{P_{\mathcal{B}_B}(k|l)}\right) \tag{9}$$

Algorithm 4 shows the steps for computing the distance  $\tilde{\mathcal{D}}$  between two networks across network portrait divergence.

---

**Algorithm 4** Compute the distance between two networks across portrait divergence.

---

input:  $G_A$  and  $G_B$

output: single value

```

Compute  $\mathcal{B}_A(G_A)$  //Return the network portrait  $\mathcal{B}$  of  $G_A$  as matrix
Compute  $\mathcal{B}_B(G_B)$  //Return the network portrait  $\mathcal{B}$  of  $G_B$  as matrix
Compute  $Q_A = P_{\mathcal{B}_A}(\mathcal{B}_A)$  //Return the Probability Distribution of  $\mathcal{B}_A$ 
Compute  $Q_B = P_{\mathcal{B}_B}(\mathcal{B}_B)$  //Return the Probability Distribution of  $\mathcal{B}_B$ 
Compute  $v_A = Reshape(Q_A)$  //Convert  $P_{\mathcal{B}_A}$  into vector
Compute  $v_B = Reshape(Q_B)$  //Convert  $P_{\mathcal{B}_B}$  into vector
return  $\tilde{\mathcal{D}}_{JS}(v_A, v_B)$  //The output is a single value
    
```

---

### 3.3.2. D-Measure

D-measure [22], a permutation-invariant and scale-adaptive approach, has been proposed to compare networks by quantifying their structures. D-measure incorporates three features related to probability distribution functions (PDFs): node distance distribution, node dispersion, and alpha centrality.

The D-measure between two given networks  $G_A$  and  $G_B$  is defined as follows [22]:

$$\tilde{\mathcal{D}}(G_A, G_B) = w_1 \sqrt{\frac{JS(v_{P_{n_A}}, v_{P_{n_B}})}{\log(2)}} + w_2 |\sqrt{NND(G_A)} - \sqrt{NND(G_B)}| + \frac{w_3}{2} \left( \sqrt{\frac{JS(P_{G_A}, P_{G_B})}{\log(2)}} + \sqrt{\frac{JS(P_{G_A^c}, P_{G_B^c})}{\log(2)}} \right) \tag{10}$$

where  $G^c$  is the complement of  $G$ , and  $w_1$ ,  $w_2$ , and  $w_3$  are arbitrary weights, such as  $w_1 + w_2 + w_3 = 1$ .

In the first term  $\sqrt{\frac{J(v_{P_{n_A}}, v_{P_{n_B}})}{\log(2)}}$ , the vectors  $v_{P_{n_A}}$  and  $v_{P_{n_B}}$  describe the node distance distributions  $P_{n_A}$  and  $P_{n_B}$  of graphs  $G_A$  and  $G_B$ , respectively. Node distance distribution  $P_n$  measures the probability that a randomly chosen pair of nodes has a shortest path of length  $d$ , such as  $P_n = p_d(i)$ , where  $\{p_d(i)\}$  is a set of nodes connected with node  $i$  at the

distance  $d$ . Then, Jensen–Shannon divergence  $JS$  is applied between the vectors  $v_{P_{n_A}}$  and  $v_{P_{n_B}}$  in order to estimate the distance.

The second term  $|\sqrt{NND(G_A)} - \sqrt{NND(G_B)}|$  measures network node dispersion by applying Jensen–Shannon divergence on node distance distribution  $v_{P_{n_A}}$  (resp.  $v_{P_{n_B}}$ ) of  $G_A$  (resp.  $G_B$ ) and normalizes it by  $\log(\text{network diameter} + 1)$ . Node dispersion ( $ND$ ) measures the distribution of nodes within a cluster  $C$  by quantifying how close the nodes are to each other, such as  $ND = \frac{\sum m_C}{n(n-1)}$  where  $m_C$  is number of edges in a cluster  $C$ .

The third term  $\sqrt{\frac{J(P_{G_A}^\alpha, P_{G_B}^\alpha)}{\log(2)}} + \sqrt{\frac{J(P_{G_A^c}^\alpha, P_{G_B^c}^\alpha)}{\log(2)}}$  extracts nodes alpha-centrality (average length of the shortest paths connecting node  $i$  with other nodes) of networks  $G_A, G_B, G_A^c$ , and  $G_B^c$ . Then, nodes' alpha-centrality values are stored into vectors  $v_{P_{G_A}^\alpha}, v_{P_{G_B}^\alpha}, v_{P_{G_A^c}^\alpha}$ , and  $v_{P_{G_B^c}^\alpha}$ . The Jensen–Shannon divergence has been used to estimate the distance between alpha-centrality vectors  $v_{P_{G_A}^\alpha}, v_{P_{G_B}^\alpha}, v_{P_{G_A^c}^\alpha}$ , and  $v_{P_{G_B^c}^\alpha}$ .

D-measure refers to the second class of comparison in [25].

Algorithm 5 shows the steps for computing the distance  $\tilde{D}$  between two networks across D-measure.

---

**Algorithm 5** Compute the distance between two networks across D-measure.

---

input:  $G_A, G_B, w_1, w_2$ , and  $w_3$  //  $w_1 = w_2 = 0.35$  and  $w_3 = 0.3$

output: single value

```

Compute  $P_{n_A}$  //Return network node distribution of  $G_A$  as matrix
Compute  $P_{n_B}$  //Return network node distribution of  $G_B$  as matrix
Compute  $v_{P_{n_A}} = \text{Reshape}(P_{n_A})$  //Convert  $P_{n_A}$  into vector
Compute  $v_{P_{n_B}} = \text{Reshape}(P_{n_B})$  //Convert  $P_{n_B}$  into vector
Compute  $\tilde{D}_{P_n}(v_{P_{n_A}}, v_{P_{n_B}})$  //Distance between  $v_{P_{n_A}}$  and  $v_{P_{n_B}}$ 
Compute  $NND_{G_A}$  //Return network node dispersion of  $G_A$  as vector
Compute  $NND_{G_B}$  //Return network node dispersion of  $G_B$  as vector
Compute  $\tilde{D}_{NND}(v_{NND_A}, v_{NND_B})$  //Distance between  $v_{NND_A}$  and  $v_{NND_B}$ 
Compute  $P_{G_A}$  //Return alpha-centrality distribution of  $G_A$  as matrix
Compute  $P_{G_B}$  //Return alpha-centrality distribution of  $G_B$  as matrix
Compute  $P_{G_A^c}^\alpha$  //Return alpha-centrality distribution of  $G_A^c$  as matrix
Compute  $P_{G_B^c}^\alpha$  //Return alpha-centrality distribution of  $G_B^c$  as matrix
Compute  $v_{P_{G_A}^\alpha} = \text{Reshape}(P_{G_A}^\alpha)$  //Convert  $P_{G_A}$  into vector
Compute  $v_{P_{G_B}^\alpha} = \text{Reshape}(P_{G_B}^\alpha)$  //Convert  $P_{G_B}$  into vector
Compute  $v_{P_{G_A^c}^\alpha} = \text{Reshape}(P_{G_A^c}^\alpha)$  //Convert  $P_{G_A^c}^\alpha$  into vector
Compute  $v_{P_{G_B^c}^\alpha} = \text{Reshape}(P_{G_B^c}^\alpha)$  //Convert  $P_{G_B^c}^\alpha$  into vector
Compute  $\tilde{D}(v_{P_{G_A}^\alpha}, v_{P_{G_B}^\alpha}) + \tilde{D}(v_{P_{G_A^c}^\alpha}, v_{P_{G_B^c}^\alpha})$  //Compute the distance between  $P_{G_A}^\alpha, P_{G_B}^\alpha, P_{G_A^c}^\alpha$ ,
and  $P_{G_B^c}^\alpha$ 
return  $\tilde{D} = w_1 \tilde{D}_{P_n}(v_{P_{n_A}}, v_{P_{n_B}}) + w_2 \tilde{D}_{NND}(v_{NND_A}, v_{NND_B}) + w_3 \tilde{D}(v_{P_{G_A}^\alpha}, v_{P_{G_B}^\alpha})$  //The out-
put is a single value

```

---

#### 4. Data

In this research, we aim to examine the effectiveness of various distance measures in identifying the similarity between movie networks and categorizing movies based on their genres. To conduct this investigation, we handpicked at least three movies from each of the following genres: horror, sci-fi, romance, and comedy. Since extracting the multilayer network from each movie script requires manual intervention, which takes much time, we limited our selection to only 15 movies presented in Table 2. To obtain movie scripts, we referred to the IMSDb database through the website at <https://imsdb.com/>.

**Table 2.** Movie Dataset.

Categories	Movies
Horror	Scream: Episode I (SC1) in 1995 Scream: Episode II (SC2) in 1997 Scream: Episode III (SC3) in 1999
Romance	Twilight: Fascination (TW1) in 2008 Twilight: New Moon (TW2) in 2009 Titanic in 1997
Comedy	500 Days of Summer in 2009 Ten Things I Hate About You in 1997 Airplane in 1979
Sci-Fi	Star Wars: A New Hope (SW1) in 1977 Star Wars: The Empire Strikes Back (SW2) in 1980 Star Wars: Return of the Jedi (SW3) in 1983 Star Wars: The Phantom Menace (SW4) in 1999 Star Wars: Attack of the Clones (SW5) in 2002 Star Wars: Revenge of the Sith (SW6) in 2005

To support our study, we had to compare our approach's outputs with ground truth data, which consist of movies ranked according to their similarities. As far as we know, no pre-existing ground truth data exist that classifies movies based on their similarity. Therefore, we had to build our ground truth data. To achieve this, we surveyed 100 participants, asking them to rank the similarity between different pairs of movies on a scale of 0 (indicating less similarity) to 10 (indicating high similarity). Table 3 shows the collected survey data, presenting the order of similarity for each pair of movies.

**Table 3.** Ground truth data.

Categories	Movies	Rank of Similarity		
		Characters	Keywords	Locations
Horror	SC1 & SC2	order 1	order 1	order 1
	SC2 & SC3	order 2	order 2	order 2
	SC1 & SC3	order 3	order 3	order 3
Romance	TW1 & TW2	order 1	order 1	order 1
	TW1 & Titanic	order 2	order 2	order 2
	TW2 & Titanic	order 2	order 2	order 2
Sci-Fi	SW5 & SW6	order 1	order 1	order 1
	SW4 & SW5	order 2	order 2	order 2
	SW4 & SW6	order 3	order 3	order 3
	SW2 & SW3	order 4	order 4	order 4
	SW1 & SW2	order 5	order 5	order 5
	SW1 & SW3	order 6	order 6	order 6
	SW3 & SW4	order 7	order 7	order 7
	SW3 & SW5	order 8	order 8	order 8
	SW2 & SW4	order 9	order 9	order 9
	SW2 & SW5	order 10	order 10	order 10
	SW1 & SW4	order 11	order 11	order 11
	SW3 & SW6	order 12	order 12	order 12
	SW1 & SW5	order 13	order 13	order 13
Comedy	Airplane & Ten Things I Hate About You	order 3	order 3	order 2
	500 Days of Summer & Ten Things I Hate About You	order 3	order 2	order 1
	Airplane & 500 Days of Summer	order 3	order 1	order 2

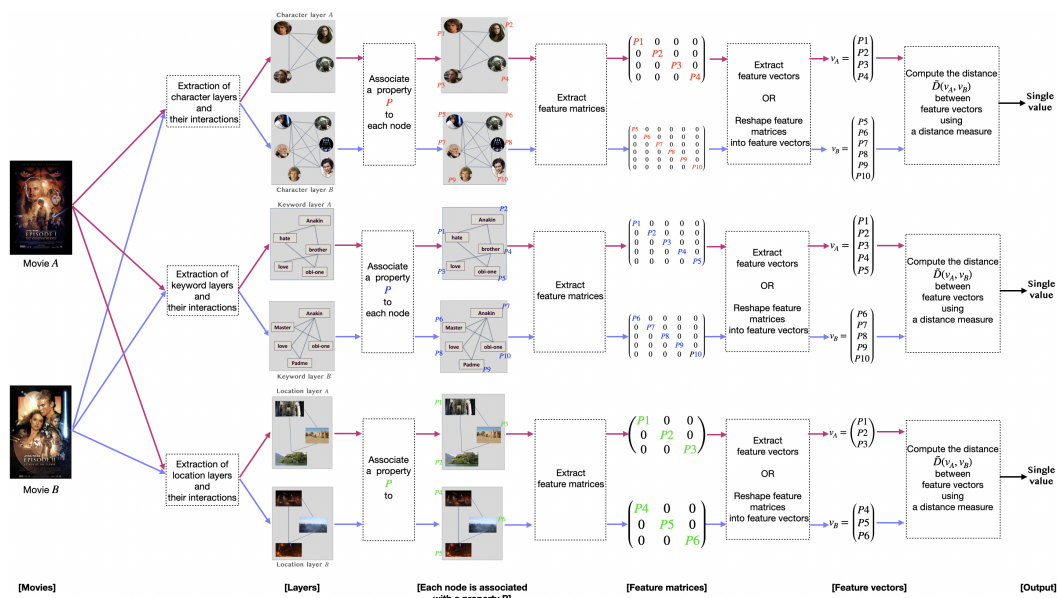
Figures A1–A3 illustrate movie networks, where every figure depicts a character, keyword, or location entity across various movie genres (sci-fi, romance, horror, and

comedy). Figure A1 enables the visualization of similarities between characters within the same genre and dissimilarities between character networks belonging to different categories. The movie networks presented in Figures A1–A3 were generated using Gephi software. For illustration, we provide movie stories in Appendix A.1.

Overall, movie network visualizations, movie stories, and survey data serve as a robust foundation for our research, leading to significant insights and valuable contributions to the field of movie analysis. Indeed, researchers can rely on the ground truth data collected in Table 3 to analyze similarities among movies and turn to the movie networks illustrated in Figures A1–A3 for visual comparisons.

### 5. Methodology

This work aims at measuring the similarity between a pair of movies. To this end, we propose a methodology composed of three main steps: (1) extracting the multilayer network from a movie script; (2) extracting the network features; (3) computing the distance between a pair of movie networks. Figure 6 shows the pipeline process.



**Figure 6.** The proposed methodology pipeline outlines the process for measuring the similarity between movies A and B. Pink denotes the process applied to movie A, while purple denotes the process applied to movie B. Firstly, after extracting movie multilayer networks of movies A and B, the distance is computed between monolayers belonging to the same entity (character layer of movie A with character layer of movie B, keywords layer of movie A with keyword layer of movie B, and location layer of movie A with movie layer of movie B). The second step involves associating features with movie networks and extracting feature matrices and vectors. Finally, the difference between feature vectors is computed using a distance measure. Then, the output is a single value that indicates the distance between movies A and B.

In the first step of our methodology, we extract a multilayer network for each movie. Section 2.2 presents, in detail, the process of multilayer extraction. At the end of this level, we obtain for each movie three layers (character, keyword, and location) and their relationships. In this work, we compare layers of the same entity, considering monolayers and intralayer links, ignoring the interlayer relationships. For example, we compare the character network of the first movie with the character network of the second movie. We provide a schema in Figure 6 to illustrate the process. The inputs consist of character layers A and B. Character layer A is associated with movie A, whereas character layer B is associated with movie B. Alternatively, the input could be a pair of keywords or location layers.

The second step consists of extracting the features of networks  $A$  and  $B$ . Network properties are crucial in network analysis as they provide us with precise information about the network's structure and characteristics. The features investigate nodes, edges, and neighborhood topology. Generally, there are two levels of network features: global and local. (i) Local features associate with each node a specific property, such as node degrees. (ii) Global features capture the overall graph, such as graph diameter. Several extraction techniques are available for extracting the global or local properties. Table 4 summarizes the features integrated into the methods that we used for our study. A vector, matrix, or single value could represent features. Figure 6 shows feature matrices extracted from movie networks  $A$  and  $B$ , respectively. Then, feature vectors  $A$  and  $B$  are extracted or reshaped from feature matrices.

The third step relies on investigating the difference between the structural features of layers  $A$  and  $B$ . Quantifying the similarity between a pair of networks involves finding the difference between their structural information. In other words, finding the distance between a pair of layers is computing the difference between their feature vectors. Once the computation is complete, a single output value is obtained, representing the distance between networks  $A$  and  $B$ .

**Table 4.** Methods and their features.

Methods	Local Feature	Global Feature	Distance
NetLSD	Permutation-invariance Scale-adaptivity	Size-invariant Scale-adaptivity	Euclidean
Laplacian spectra	✗	Eigenvalue spectrum of the Laplacian matrix	Euclidean
NetMF	Random walks	✗	Euclidean
D-measure	Node dispersion	Node Alpha centrality	distance distribution Jensen-Shannon
Network portrait divergence	✗	Node degree Shortest path Next-nearest neighbors distribution	degree length distribution Jensen-Shannon

If  $A$  and  $B$  have identical structural information, the distance between their feature vectors should be 0. The more the output value approaches 0, the higher the similarity between networks  $A$  and  $B$ . On the other hand, the further away the output is from 0 and the closer to 1, the more  $A$  and  $B$  are dissimilar.

Note that all the distance measures used in this study follow steps 2 and 3. We selected a set of approaches that calculates the distance between a pair of networks (Section 3). Then, we applied the measures to the movie networks. To provide an overview of the distance between a set of movies, we present the output values in a heat map, as shown in Section 6.

## 6. Experimental Evaluation

In this section, we discuss and analyze the results obtained. Table 5 illustrates the performance of different approaches in comparing character layers of horror, romance, sci-fi, and comedy movies. Idem, Table 7 shows the distance between the keyword layers, while Table 9 displays the distance between the location layers. We summarize in Tables 6, 8, and 10 the performance of the approaches in estimating the distance between the layers in different categories. Umap in Figures 7–9 displays the classification of characters, keyword, and location networks across various movie genres by applying the five distance measures.

### 6.1. What Is the Best Measure for Comparing Horror Movies?

According to the ground truth data, the most similar chapters in the Scream Saga are I and II. Episodes II and III are on the second level. Then, episodes I and III are on the third level. Regarding the character layer divergence in Table 5, we notice that NetLSD is the unique measure verifying the order of similarity of movies as the ground truth data. Conversely, NetMF detected a high similarity between episodes I and II as the ground truth data. However, it failed to determine the proper order of similarity for the other chapters.

Network portrait divergence, Laplacian spectra, and D-measure cannot improve the correct order of similarity of episodes according to the ground truth data. Indeed, they detected a high similarity between episodes II and III and less similarity between episodes I and II. In addition, the network portrait divergence outputs approximate values comparing chapters I and II ( $D_{JS} = 0.98$ ) and I and III ( $D_{JS} = 0.97$ ). In other words, the network portrait divergence predicts the same order of similarity of episode I with episodes II and III. Despite that, 0.97 and 0.98 are too far from 0, which means a high dissimilarity between episodes. All in all, *NetLSD* is the best measure for comparing character layers in horror movies.

Regarding Table 7, Laplacian spectra, *NetLSD*, network portrait divergence, and D-measure show high similarity between episodes I and II of the Scream Saga. Indeed, according to the ground truth data in Table 3, episodes I and II are the most similar. As shown in Table 7, the Laplacian spectra ranks the similarity between Scream Saga chapters in the same order as presented in the ground truth data. In opposition to the other measures, they ordered episodes I and III to be more similar to those II and III. As a result, they did not classify the movies in the proper order. However, episode III turns around *Stab*, a movie parody of episode I. Maybe these measures find a high similarity between both episodes. Therefore, *Laplacian spectra* is the best measure for comparing the keyword layers of horror movies.

As shown in the ground truth data in Table 3, episodes I and II are more similar than episodes II and III, and episodes II and III are more comparable than episodes I and III. Conversely, most scenes in the Scream Saga take place in houses, gardens, streets, and schools. Accordingly, the similarity of locations between the three episodes is about 90%. *NetLSD* is the unique measure ranking the similarity between location layers in the proper order, as shown in the ground truth data. Thus, *NetLSD* outperforms the other approaches (Table 9). In brief, *NetLSD* is the best measure for estimating the distance between location layers in horror movies.

## 6.2. What Is the Best Measure for Comparing Romance Movies?

According to the ground truth data in Table 3, the Twilight Saga episodes I and II share the most similarities, whereas Titanic and the Twilight episodes share the least. When comparing the romance character layers in Table 3, only the network portrait divergence outputs a high similarity between episode I and episode II of Twilight, as shown in the ground truth data. In opposition, *NetMF* and D-measure rank Titanic and the first chapter of Twilight in the first order. On the other hand, Laplacian spectra and *NetLSD* rank Titanic and the second chapter of Twilight in the first order. Thus, *NetLSD*, *NetMF*, network Laplacian, and D-measure did not perform well in comparing character layers of romance movies. In brief, the *network portrait divergence* seems to be the best measure for comparing character layers in romance movies.

When comparing keyword relationships in romance movies, the network portrait divergence, the D-measure, and the *NetMF* assume high similarities between Titanic and episode II of Twilight. The *NetLSD*, on the other hand, finds a high similarity between Titanic and episode I of Twilight. According to the ground truth data, episode I and episode II of Twilight are the most similar. However, love is the play's dominant theme and the most important in romance movies. Thus, there are common keywords between the Twilight and Titanic stories. *NetLSD* reveals high similarity between Titanic and episode I of Twilight, but it also shows a high similarity between Titanic and episode II of Twilight. Because of this ambiguity, we cannot consider *NetLSD* as an appropriate metric for comparing keyword layers in romance films. In summary, no method effectively compares keyword layers in romance films.

According to the ground truth data (Table 3), the similarity between Twilight episodes is in the first rank, while the similarities between Titanic and Twilight chapters are in the second order. Regarding the results in Table 9, the Laplacian spectra is the unique measure that detected the similarity between location layers of the romance movies in the same order as the ground truth data. Indeed, the Laplacian spectra classed Twilight chapters in the first order with a distance of 3.74, whereas it classed the similarities between Twilight chapters and Titanic in the second order with a value of 18.82. In brief, the *Laplacian spectra* is the proper measure for comparing location layers in romance movies.



### 6.3. What Is the Best Measure for Comparing Sci-Fi Movies?

Regarding the ground truth data, the Star Wars Saga's episodes V and VI are the most similar (order 1), followed by episodes IV and V (order 2), and episodes I and VI are the least similar (order 15). The network portrait divergence detected that episodes V and VI are the most similar, and episodes I and VI are the least, as shown in the ground truth data, but it could not reveal the proper order for the other episodes. However, it finds that some episodes are more similar than others, such as episodes III and IV being more similar than III and V, episodes I and II being more alike than I and III, and episodes II and V being more similar than II and VI. D-measure also reveals that episodes V and VI are the most similar, and episodes I and VI are the least. Furthermore, D-measure ranked episodes II and VI in the same order as episodes I and VI. That is because it outputs a distance of 0.25 between each of them. Indeed, in the ground truth data, episodes II and VI are placed in the order 14 just before episodes I and VI. On the other hand, D-measure placed some episodes in the same order, such as episodes I and II with episodes III and IV, and episodes II and IV with episodes I and IV. Regarding the ground truth data, those episodes are placed near each other. The other approaches did not reveal the proper order of episodes or at least return the higher and lower distance as the ground truth data. In brief, the *network portrait divergence* and D-measure are the best measures for comparing characters in sci-fi movies.

All of the measures did not rank the keyword layers in the correct order. Furthermore, they show a very high dissimilarity between episodes. Except for the D-measure, the distance between movies does not surpass 0.3. In conclusion, no measure was selected to be the most effective to compare keyword layers in sci-fi movies.

Similar to keyword layers, no measure orders the similarity between location networks of Star Wars movies in the same order as the ground truth data. However, the D-measure shows less dissimilarity between episodes (0.2), while NetMF shows less similarity (0.59). In brief, no measure can reveal sci-fi movies' most similar location layers.

### 6.4. What Is the Best Measure for Comparing Comedy Movies?

According to the ground truth data in comedy movies, similarities between characters are in the third order. That explains the high difference between characters and their relationships in the three films: *Airplane*, *Ten Things I Hate About You*, and *500 Days of Summer*. The network portrait divergence shows high distances between the three movies. That is, it outputs a distance of 0.81 between *Airplane* and *Ten Things I Hate About You*, a distance of 0.90 between *Airplane* and *500 Days of Summer*, and a value of 0.94 between *500 Days of Summer* and *Ten Things I Hate About You*. The values are far from 0 and near to 1, which justifies the high divergence between the character layers. Likewise, Laplacian spectra and NetMF show high distances through the three movies. In opposition, D-measure and NetLSD show close distances between character layers. In summary, the *network portrait divergence*, D-measure, and NetLSD seem to be proper measures for comparing character relationships in comedy movies.

Regarding the ground truth data in keyword layers, the similarity between *Airplane* and *500 Days of Summer* is in the first rank, followed by *500 Days of Summer* and *Ten Things I Hate About You*, then *Airplane* and *Ten Things I Hate About You*. The network portrait divergence and the Laplacian spectra perform exceptionally well in comparing keyword layer relationships. Indeed, they ranked the movies in the same order as the ground truth data. In opposition, NetMF, NetLSD, and D-measure did not find the correct similarity between keyword layers. In conclusion, *network portrait divergence* and the *Laplacian spectra* are the proper approaches for comparing keyword relationships in comedy movies.

The ground truth data show a high similarity between location layers of the movies *500 Days of Summer* and *Ten Things I Hate About You*, followed by *Airplane* and *Ten Things I Hate About You*, and *Airplane* and *500 Days of Summer* in the second order. The approaches NetMF and NetLSD reveal the high similarity between *500 Days of Summer* and *Ten Things I Hate About You*, but they did not find the proper order for the other layers. NetLSD outputs a value of 6.74 comparing the movies *Airplane* and *500 Days of Summer*, and a value of 7.06 for

the movies *Airplane* and *Ten Things I Hate About You*. However, the interval distance between both values is not far. In opposition to NetMF, it finds incomparable distances: 5.56 between the movies *Airplane* and *500 Days of Summer*, and 17.12 between *Airplane* and *Ten Things I Hate About You*. D-measure shows a high similarity between *Airplane* and both movies *500 Days of Summer* and *Ten Things I Hate About You*, and a lower similarity between *500 Days of Summer* and *Ten Things I Hate About You*. Thus, we can consider *NetLSD* as a proper measure for comparing the relationship between location layers of the comedy movies.

### 6.5. What Is the Best Measure for Measuring the Similarity between Character Layers?

From Table 6, the network portrait divergence outperforms the other approaches in comparing the relationship between characters in the romance and sci-fi categories. Furthermore, it gives good results comparing comedy movies, but it cannot analyze character relationships in the horror category. On the other hand, the *NetLSD* is a good measure for comparing the horror category. Moreover, *NetLSD* and D-measure can compare character relationships in comedy movies. However, they cannot give good results in analyzing the other genres. Laplacian spectra and NetMF can not reveal proper relationships between character layers in opposition. In brief, the network portrait divergence would be a good measure for comparing character layers if it could compare horror movies. However, we can select the network portrait divergence as a proper measure for comparing romance, sci-fi, and comedy movies. Then, we choose *NetLSD* as a good measure for comparing horror movies.

**Table 5.** The distance between character layers using Laplacian spectra, network portrait divergence, *NetLSD*, NetMF, and D-measure. Distance values are scaled between 0 and 1. Bold text indicates the most similar movies within a genre. In the Laplacian spectra, *NetLSD*, and NetMF columns, values are normalized by dividing each value by the maximum value in its corresponding column.

Categories	Movies	Methods				
		Laplacian Spectra Distance	Network Portrait Divergence	NetLSD	NetMF	D-Measure
Horror	SC1 & SC2	192.60 (0.99)	0.98	<b>2.27</b> (0.45)	<b>3.50</b> (0.18)	0.67
	SC2 & SC3	<b>90.15</b> (0.46)	<b>0.890</b>	2.34 (0.46)	11.42 (0.6)	<b>0.31</b>
	SC1 & SC3	177.85 (0.92)	0.97	4.60 (0.92)	8.45 (0.44)	0.44
Romance	TW1 & TW2	11.90 (0.06)	<b>0.91</b>	3.18 (0.63)	15.28 (0.8)	0.13
	TW1 & Titanic	11.14 (0.05)	0.96	4.88 (0.97)	<b>11.68</b> (0.61)	<b>0.07</b>
	TW2 & Titanic	<b>5.73</b> (0.02)	0.98	<b>1.70</b> (0.34)	14.70 (0.77)	0.12
Sci-Fi	SW5 & SW6	21.25 (0.11)	<b>0.13</b>	0.65 (0.13)	15.56 (0.82)	<b>0.06</b>
	SW4 & SW5	27.75 (0.14)	0.19	<b>0.04</b> (0.008)	14.82 (0.78)	0.1
	SW4 & SW6	<b>13.25</b> (0.06)	0.22	0.69 (0.14)	14.23 (0.74)	0.12
	SW2 & SW3	30.53 (0.15)	0.27	0.62 (0.12)	16.55 (0.87)	0.13
	SW1 & SW2	46.10 (0.23)	0.28	0.42 (0.08)	15.46 (0.81)	0.07
	SW1 & SW3	23.84 (0.12)	0.30	1.04 (0.2)	16.88 (0.89)	0.11
	SW3 & SW4	36.59 (0.18)	<b>0.13</b>	0.19 (0.4)	18.08 (0.95)	0.07
	SW3 & SW5	27.00 (0.13)	0.15	0.23 (0.05)	14.80 (0.78)	0.16
	SW2 & SW4	46.51 (0.24)	0.26	0.80 (0.16)	16.53 (0.87)	0.15
	SW2 & SW5	29.90 (0.15)	0.32	0.84 (0.17)	15.17 (0.79)	0.23
	SW1 & SW4	43.10 (0.22)	0.34	1.22 (0.24)	18.01 (0.94)	0.15
	SW3 & SW6	29.36 (0.15)	0.21	0.88 (0.18)	14.32 (0.75)	0.18
	SW1 & SW5	35.92 (0.18)	0.37	1.26 (0.25)	13.50 (0.71)	0.22
SW2 & SW6	28.00 (0.14)	0.33	1.50 (0.3)	15.19 (0.8)	0.25	
SW1 & SW6	41.67 (0.21)	0.39	1.92 (0.38)	<b>13.30</b> (0.7)	0.25	
Comedy	Airplane & 10 Things I Hate About You	<b>64.30</b> (0.33)	<b>0.81</b>	2.26 (0.45)	16.35 (0.86)	<b>0.12</b>
	500 Days of Summer & 10 Things I Hate About You	72.90 (0.37)	0.94	<b>0.10</b> (0.02)	<b>11.18</b> (0.59)	0.25
	Airplane & 500 Days of Summer	80.59 (0.41)	0.90	2.16 (0.43)	12.59 (0.66)	0.30

**Table 6.** A comprehensive character checklist table: Evaluating measures in revealing the similarity between character networks from different movie genres with checkmarks.

Type of Methods	Measures	Horror	Romance	Sci-Fi	Comedy
Spectral	NetLSD	✓	✗	✗	✓
	Laplacian Spectra	✗	✗	✗	✗
Embedding	NetMF	✗	✗	✗	✗
Statistical	D-measure	✗	✗	✓	✓
	Network Portrait Divergence	✗	✓	✓	✓

### 6.6. What Is the Best Measure for Measuring the Similarity between Keyword Layers?

From Table 8, no approach seems to be a proper choice for comparing keyword layers in four categories. The Laplacian spectra gives a good result in comparing keywords in horror and comedy categories, but it fails to analyze keywords in romance and sci-fi categories. On the other hand, the network portrait divergence can only compare comedy movies. However, we can select the Laplacian spectra as a proper measure to compare keyword relationships in horror and comedy movies.

**Table 7.** The distance between keyword layers using Laplacian spectra, network portrait divergence, NetLSD, NetMF, and D-measure. Distance values are scaled between 0 and 1. Bold text indicates the most similar movies within a genre. In the Laplacian spectra, NetLSD, and NetMF columns, values are normalized by dividing each value by the maximum value in its corresponding column.

Categories	Movies	Methods				
		Laplacian Spectra	Network Portrait Divergence	NetLSD	NetMF	D-Measure
Horror	SC1 & SC2	<b>2.68</b> (0.006)	<b>0.23</b>	<b>1.20</b> (0.15)	25.91 (0.63)	<b>0.10</b>
	SC2 & SC3	4.68 (0.01)	0.89	7.99 (0.99)	<b>24.42</b> (0.6)	0.69
	SC1 & SC3	9.43 (0.02)	0.81	6.79 (0.85)	25.27 (0.61)	0.61
Romance	TW1 & TW2	<b>11.14</b> (0.02)	0.69	3.75 (0.47)	29.99 (0.73)	0.43
	TW1 & Titanic	11.89 (0.03)	0.71	<b>0.78</b> (0.09)	29.72 (0.72)	0.47
	TW2 & Titanic	<b>11.14</b> (0.02)	<b>0.44</b>	2.97 (0.38)	<b>28.84</b> (0.7)	<b>0.12</b>
Sci-Fi	SW5 & SW6	200.77 (0.52)	0.72	1.13 (0.14)	37.50 (0.91)	0.26
	SW4 & SW5	136.05 (0.35)	0.46	0.64 (0.08)	39.13 (0.95)	0.14
	SW4 & SW6	142.04 (0.37)	0.54	0.48 (0.06)	36.49 (0.89)	0.15
	SW2 & SW3	139.77 (0.36)	<b>0.29</b>	0.06 (0.007)	32.57 (0.79)	<b>0.04</b>
	SW1 & SW2	271.94 (0.7)	0.61	2.00 (0.25)	35.55 (0.87)	0.12
	SW1 & SW3	377.72 (0.98)	0.61	2.06 (0.26)	36.19 (0.88)	0.11
	SW3 & SW4	124.80 (0.32)	0.65	1.91 (0.23)	37.62 (0.92)	0.16
	SW3 & SW5	<b>102.60</b> (0.27)	0.81	2.55 (0.31)	35.64 (0.87)	0.28
	SW2 & SW4	125.14 (0.32)	0.68	1.86 (0.23)	37.18 (0.76)	0.18
	SW2 & SW5	211.24 (0.55)	0.82	2.49 (0.31)	34.56 (0.84)	0.30
	SW1 & SW4	316.06 (0.82)	0.41	<b>0.15</b> (0.02)	40.05 (0.98)	0.11
	SW3 & SW6	203.02 (0.53)	0.56	1.43 (0.17)	32.96 (0.8)	0.06
	SW1 & SW5	383.35 (0.99)	0.66	0.51 (0.06)	38.49 (0.94)	0.23
	SW2 & SW6	142.34 (0.37)	0.59	1.37 (0.17)	<b>31.56</b> (0.76)	0.06
SW1 & SW6	213.98 (0.55)	0.33	0.63 (0.08)	35.68 (0.87)	0.09	
Comedy	Airplane & 10 Things I Hate About You	17.05 (0.04)	0.56	<b>2.12</b> (0.26)	29.30 (0.71)	0.19
	500 Days of Summer & 10 Things I Hate About You	14.67 (0.03)	0.47	6.58 (0.82)	<b>28.47</b> (0.7)	<b>0.09</b>
	Airplane & 500 Days of Summer	<b>12.73</b> (0.03)	<b>0.43</b>	4.46 (0.56)	29.63 (0.72)	0.15

**Table 8.** A comprehensive keyword checklist table: Evaluating measures in revealing the similarity between keyword networks from different movie genres with checkmarks.

Type of Methods	Measures	Horror	Romance	Sci-Fi	Comedy
Spectral	NetLSD	✗	✗	✗	✗
	Laplacian Spectra	✓	✗	✗	✓
Embedding	NetMF	✗	✗	✗	✗
Statistical	D-measure	✗	✗	✗	✗
	Network Portrait Divergence	✗	✗	✗	✓

### 6.7. What Is the Best Measure for Measuring the Similarity between Location Layers?

From Table 10, no approach seems to be a proper choice for comparing keyword layers in four categories. The NetLSD gives a good result in comparing location layers in horror and comedy categories, but it fails to analyze locations in romance and sci-fi categories. On the other hand, the Laplacian spectra can only compare the similarity through romance movies. However, we can select the NetLSD as a proper measure to compare location relationships in horror and comedy movies, and choose the Laplacian spectra as a measure to analyze location layers in the romance category.

**Table 9.** The distance between location layers using Laplacian spectra, network portrait divergence, NetLSD, NetMF, and D-measure. Distance values are scaled between 0 and 1. Bold text indicates the most similar movies within a genre. In the Laplacian spectra, NetLSD, and NetMF columns, values are normalized by dividing each value by the maximum value in its corresponding column.

Categories	Movies	Methods				
		Laplacian Spectra	Network Portrait Divergence	NetLSD	NetMF	D-Measure
Horror	SC1 & SC2	5.76 (0.11)	0.87	<b>3.57</b> (0.39)	17.74 (0.55)	0.65
	SC2 & SC3	<b>5.03</b> (0.1)	<b>0.33</b>	5.02 (0.56)	16.23 (0.5)	<b>0.12</b>
	SC1 & SC3	5.46 (0.1)	0.85	8.59 (0.95)	<b>15.73</b> (0.49)	0.67
Romance	TW1 & TW2	<b>3.74</b> (0.07)	0.59	0.44 (0.05)	24.16 (0.75)	0.35
	TW1 & Titanic	18.82 (0.38)	0.62	0.46 (0.05)	<b>19.74</b> (0.62)	0.31
	TW2 & Titanic	18.82 (0.38)	<b>0.37</b>	<b>0.05</b> (0.005)	20.56 (0.64)	<b>0.19</b>
Sci-Fi	SW5 & SW6	22.08 (0.44)	0.08	1.16 (0.13)	20.56 (0.64)	0.12
	SW4 & SW5	49.91 (0.99)	0.23	0.76 (0.08)	31.64 (0.98)	0.09
	SW4 & SW6	18.21 (0.36)	0.26	1.92 (0.21)	21.85 (0.68)	0.16
	SW2 & SW3	8.05 (0.16)	<b>0.00</b>	0.33 (0.04)	23.24 (0.73)	0.07
	SW1 & SW2	5.80 (0.12)	0.01	<b>0.20</b> (0.02)	25.22 (0.78)	0.08
	SW1 & SW3	13.73 (0.27)	0.02	0.53 (0.06)	23.37 (0.73)	0.07
	SW3 & SW4	10.34 (0.2)	0.58	2.91 (0.32)	28.50 (0.89)	0.18
	SW3 & SW5	9.73 (0.19)	0.30	2.15 (0.23)	29.38 (0.92)	0.15
	SW2 & SW4	<b>5.60</b> (0.11)	0.55	2.58 (0.28)	27.30 (0.85)	0.20
	SW2 & SW5	10.90 (0.21)	0.26	1.82 (0.2)	25.91 (0.8)	0.18
	SW1 & SW4	10.35 (0.2)	0.50	2.38 (0.2)	27.44 (0.85)	0.16
	SW3 & SW6	19.71 (0.39)	0.15	0.99 (0.11)	<b>18.93</b> (0.59)	0.11
	SW1 & SW5	16.55 (0.33)	0.20	1.61 (0.18)	26.21 (0.81)	0.12
	SW2 & SW6	16.75 (0.33)	0.13	0.66 (0.07)	19.96 (0.62)	0.13
SW1 & SW6	15.40 (0.31)	0.10	0.46 (0.05)	20.13 (0.62)	<b>0.06</b>	
Comedy	Airplane & 10 Things I Hate About You	15.45 (0.3)	<b>0.57</b>	7.06 (0.78)	17.12 (0.53)	<b>0.25</b>
	500 Days of Summer & 10 Things I Hate About You	11.14 (0.22)	0.61	<b>0.33</b> (0.03)	<b>5.20</b> (0.16)	0.32
	Airplane & 500 Days of Summer	<b>10.80</b> (0.22)	0.67	6.74 (0.75)	5.56 (0.17)	<b>0.25</b>

**Table 10.** A comprehensive keyword checklist table: Evaluating measures in revealing the similarity between keyword networks from different movie genres with checkmarks.

Type of Methods	Measures	Horror	Romance	Sci-Fi	Comedy
Spectral	NetLSD	✓	✗	✗	✓
	Laplacian Spectra	✗	✓	✗	✗
Embedding	NetMF	✗	✗	✗	✗
Statistical	D-measure	✗	✗	✗	✗
	Network Portrait Divergence	✗	✗	✗	✗

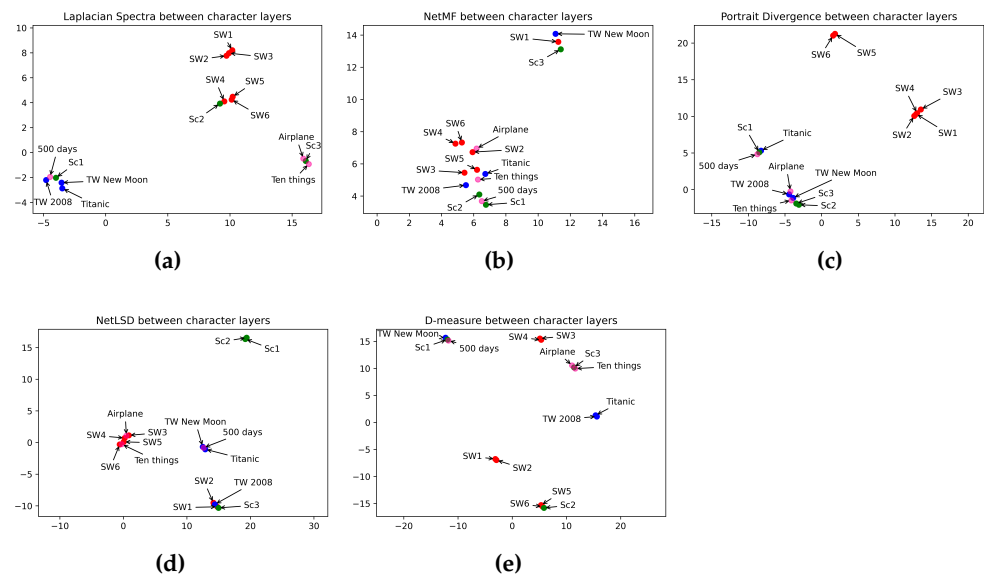
### 6.8. What Is the Best Measure for Comparing the Similarity between Movies from Different Categories?

In this section, we compare the similarity between film genres. We applied five graph distance measures to the character, keyword, and location layers of movies from different genres. Then we visualized, using Umap, the performance of distance measures in classifying movie genres. Figures 7–9 show the interpretation of graph distance measures in categorizing character, keyword, and location networks from different movie genres.

Comparing character layers in Figure 7, NetMF (Figure 7b), and D-measure (Figure 7e) failed to detect the dissimilarity between character layers in different movie genres. That is because NetMF mapped almost all movies from different categories closed to each other, and D-measure mapped movies from the same genre so far from each other. The Laplacian spectra (Figure 7a) grouped episodes I, II, and III in one space and episodes IV, V, and VI together, preserving a close distance between the two groups. Indeed, the Star Wars Saga consists of two trilogies: prequel (episodes I, II, and III) and sequel (episodes IV, V, and VI). Laplacian spectra placed episode III of Scream closer to two comedy movies. That is because episode III of Scream has a comedy aspect, too. The network portrait divergence (Figure 7c) embedded *Ten Things I Hate About You* and *Airplane* in the same space as episodes I and II of Twilight. Indeed, the movies *Ten Things I Hate About You* and *Airplane* are comedies, but they have a romantic side. The network portrait divergence placed episodes I, II, III, and IV in the same space. However, it kept episodes V and VI far from them. Also, the network portrait divergence placed episodes I and II of Scream close to romance and comedy movies, while they are not similar. NetLSD (Figure 7d) shows a high similarity between the four movies from the sci-fi genre. Indeed, it embedded the four sci-fi movies at a close distance. Also, NetLSD mapped two horror movies at a close distance. Idem for romance and comedy movies. NetLSD classifies horror movie genres in a high space from comedy movies. Also, it showed a far distance between the four sci-fi movies, the two romance movies, and the two horror movies. In opposition, it mapped two comedy movies and four sci-fi movies simultaneously. As *airplane* and "Ten Things I Hate About You" do not belong to the sci-fi category, NetLSD failed to classify the comedy movie genre. Furthermore, it showed a high similarity between one movie in horror, romance, and sci-fi genres. However, observing the performance of the other measures, NetLSD attained just a few errors in classifying character networks by category. Table 11 summarizes how distance measures perform in categorizing movie genres using character networks.

Regarding Figure 8, we observe the efficiency of the Laplacian spectra (Figure 8a) in embedding all the sci-fi movies in the same class, preserving a close distance between the prequel and sequel trilogies. Furthermore, it detects a high difference between sci-fi movies and other film genres. That is because it mapped sci-fi movies in a far distance from the others. The Laplacian spectra embedded episodes I and II of the Scream Saga close to each other and far from the remaining movie genres. Furthermore, it embedded romance movies close to each other. However, the Laplacian spectra placed episode III of Scream close to romance movies even though this episode does not tell a love story. The network portrait divergence (in Figure 8c) outperforms other measures in classifying the comedy genre. That is because it embedded the three comedies in the same space. The

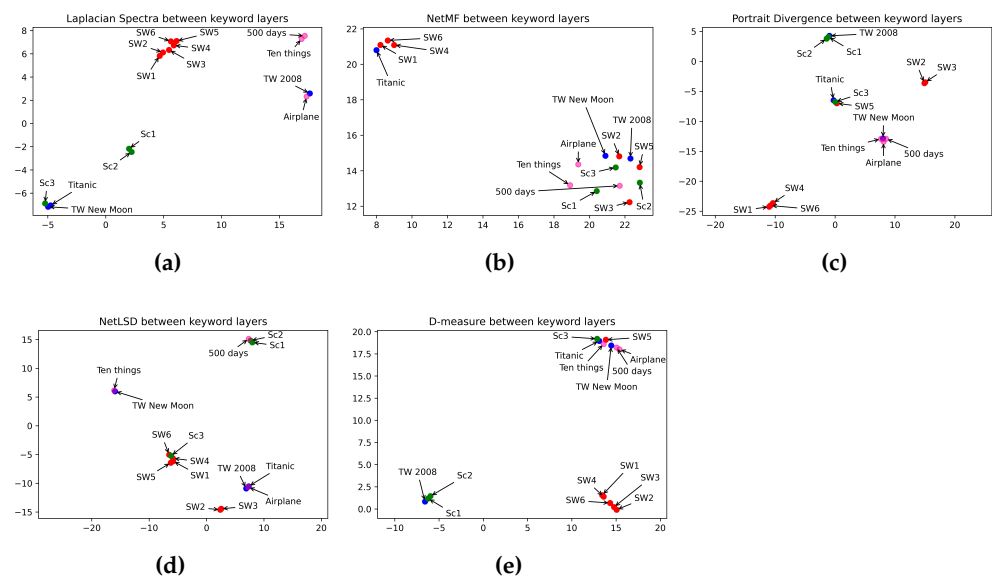
network portrait divergence reveals a high distance between comedy movies and other movie categories. However, the network portrait divergence showed an error in mapping one romance movie with comedy movies. Furthermore, it failed in embedding romance movies at a high distance from horror movies. Again, the NetMF (Figure 8b) failed to detect the dissimilarity between movie genres. In opposition, D-measure (Figure 8e) mapped five sci-fi movies far from other movie categories. All of the measures, excluding NetMF, placed episodes I and II of the Scream Saga close to each other and far from episode III. That is because episode III of Scream has fewer crimes than episodes I and II on one side, and episode III has an aspect of the comic on another side. Table 12 summarizes how distance measures perform in categorizing movie genres using keyword networks.



**Figure 7.** Visualization of 5 graph distance measures applied to character movie networks from sci-fi, romance, horror, and comedy genres. (a) Laplacian spectra. (b) NetMF. (c) Portrait divergence. (d) NetLSD. (e) D-measure. Similar movies are grouped at a close point in space, while dissimilar movies appear farther apart. Each color represents a movie genre, which makes it easy to visualize the performance of distance measures in grouping movies belonging to the same genre: red for sci-fi, blue for romance, green for horror, and pink for comedy.

**Table 11.** Table of character-based movie classification checklist.

Measures	Horror	Romance	Sci-fi	Comedy	Horror	Horror	Horror	Romance	Romance	Sci-fi
					vs. Romance	vs. Sci-fi	vs. Comedy	vs. Sci-fi	vs. Comedy	vs. Comedy
NetLSD	✓	✓	✓	✓	✓	✓	✓	✓	✓	✗
Laplacian Spectra	✗	✓	✓	✓	✗	✗	✓	✓	✓	✓
NetMF	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗
D-measure	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗
Network Portrait Divergence	✓	✓	✓	✓	✗	✓	✗	✓	✓	✓

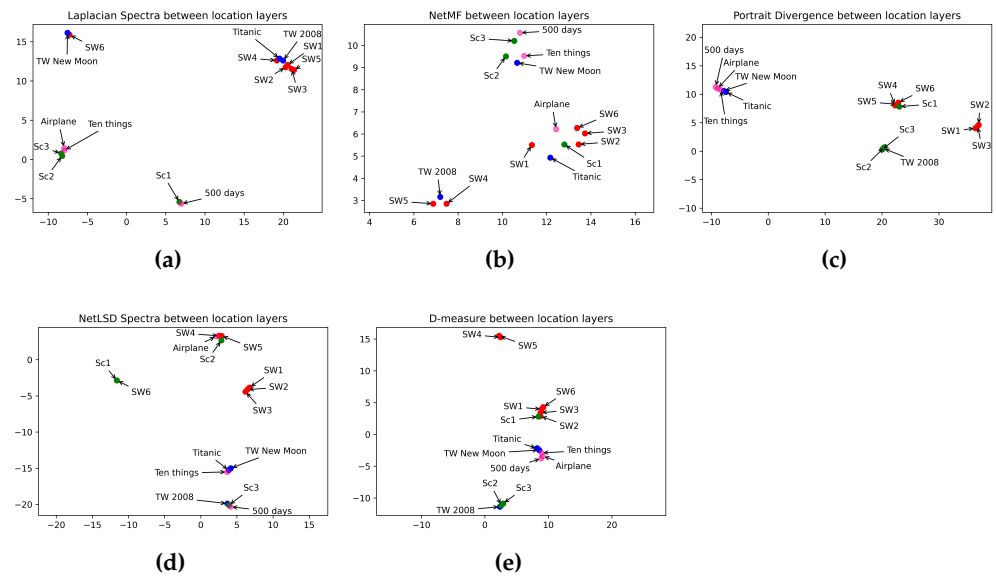


**Figure 8.** Visualization of 5 graph distance measures applied to keyword movie networks from sci-fi, romance, horror, and comedy genres. (a) Laplacian spectra. (b) NetMF. (c) Portrait divergence. (d) NetLSD. (e) D-measure. Similar movies are grouped at a close point in space, while dissimilar movies appear farther apart. Each color represents a movie genre, which makes it easy to visualize the performance of distance measures in grouping movies belonging to the same genre: red for sci-fi, blue for romance, green for horror, and pink for comedy.

**Table 12.** Table of keyword-based movie classification checklist

Measures	Horror	Romance	Sci-fi	Comedy	Horror	Horror	Horror	Romance	Romance	Sci-fi
					vs. Romance	vs. Sci-fi	vs. Comedy	vs. Sci-fi	vs. Comedy	vs. Comedy
NetLSD	✓	✓	✓	✗	✗	✗	✗	✓	✗	✓
Laplacian Spectra	✓	✓	✓	✓	✗	✓	✓	✓	✓	✓
NetMF	✓	✓	✗	✓	✗	✗	✗	✗	✗	✗
D-measure	✓	✗	✓	✓	✗	✓	✓	✓	✗	✓
Network Portrait Divergence	✓	✗	✗	✓	✗	✓	✓	✗	✗	✓

Observing location layers in Figure 9, the Laplacian spectra (Figure 9a) mapped sci-fi and romance movies at a high distance from comedy and horror movies. But, it showed a strong connection between sci-fi and romantic films on one side and horror and comedy movies on the other. Thus, the Laplacian spectra failed to reveal the difference between horror and comedy genres. Idem for romance and sci-fi genres. NetMF (Figure 9b) still failed in categorizing movies. D-measure (Figure 9e) plotted comedy movies in the same spot. The network portrait divergence (Figure 9c), again, performed well in classifying comedy movies. Indeed, the network portrait divergence embedded the three comedy genres in the same place. Furthermore, it revealed a high distance when comparing comedy movies to horror and sci-fi genres. The network portrait divergence grouped the prequel trilogy of the Star Wars Saga in one space and the sequel trilogy in another, preserving a close distance between both trilogies. The network portrait divergence mapped comedy movies closer to two romance movies. Note that *Ten Things I Hate About You* and *Airplane* have a romance aspect, too. Table 13 summarizes how distance measures perform in categorizing movie genres using location networks.



**Figure 9.** Visualization of 5 graph distance measures applied to location movie networks from sci-fi, romance, horror, and comedy genres. (a) Laplacian spectra. (b) NetMF. (c) Portrait divergence. (d) NetLSD. (e) D-measure. Similar movies are grouped at a close point in space, while dissimilar movies appear farther apart. Each color represents a movie genre, which makes it easy to visualize the performance of distance measures in grouping movies belonging to the same genre: red for sci-fi, blue for romance, green for horror, and pink for comedy.

**Table 13.** Table of location layers-based movie classification checklist.

Measures	Horror	Romance	Sci-fi	Comedy	Horror	Horror	Horror	Romance	Romance	Sci-fi
					vs. Romance	vs. Sci-fi	vs. Comedy	vs. Sci-fi	vs. Comedy	vs. Comedy
NetLSD	✗	✓	✗	✗	✗	✗	✗	✓	✗	✗
Laplacian Spectra	✓	✓	✓	✓	✓	✓	✗	✗	✓	✓
NetMF	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗
D-measure	✗	✗	✓	✓	✗	✓	✓	✓	✗	✓
Network Portrait Divergence	✓	✗	✓	✓	✓	✗	✓	✓	✓	✓

### 7. Discussion and Conclusions

The impact of network features on similarities between movie networks in terms of their entities and structures is an interesting area of research. Depending on a network entity or a movie category, a measure may be performant. Understanding network patterns can provide valuable insights into the entities and structures present within these networks. According to our previous study [26], character layers are small-world networks, keyword layers are typically scale-free networks, and location layers are chain-type networks. This observation highlights the importance of exploring network patterns to better understand movie networks and their underlying structures.

When comparing movies of the same genre, the network portrait divergence is a reliable approach for analyzing character layers of comedy, romance, and sci-fi. However, it is not suitable for comparing characters in horror movies. Instead, spectral methods are more appropriate for measuring similarity within the horror genre. In particular, the NetLSD method outperformed other approaches in analyzing character and location layers, while the Laplacian spectra method was superior in comparing keyword layers. D-measure revealed the correct order between character layers in sci-fi and comedy genres.



Despite not ranking the remaining movies and network entities in the same order as the dataset, D-measure showed a high similarity between films of the same genre for sci-fi, romance, and comedy. The embedding approach, NetMF, was not a performant measure as it did not perform any movie genre or network entity. Neither measure gave good results in comparing keyword and location layers in sci-fi movies. Furthermore, none of the approaches produced efficient results when comparing keywords of the romance category. However, the Laplacian spectra was the unique measure that performed better in investigating location layers of romance movies.

Regarding movie genre classification, NetLSD performed well in classifying character layers across all movie genres, except for distinguishing between sci-fi and comedy networks. The network portrait divergence accurately categorized overall movies based on their character and location layers, with few exceptions. The Laplacian spectra outperformed the other measures in classifying movies through keyword and location layers. Again, NetMF and D-measure provided ambiguous results. In brief, the Laplacian spectra was the most effective method for classifying movie genres and identifying similarities and differences between movies based on keyword and location networks. In scale-free networks, most nodes follow power-law (nodes have very few connections) distribution for their degree, while only a few nodes form hubs (a small number of highly connected nodes). This network structure generates a unique eigenvalue pattern in their Laplacian spectra, where the eigenvalues associated with the low-degree nodes tend to be negligible, and the eigenvalues associated with hubs are significantly larger. Nodes in chain-type networks are often connected without forming loops, giving them a linear and acyclic structure. The eigenvalues of Laplacian spectra can provide valuable insights into the interconnectivity between nodes and the chain length. Small-world networks exhibit two main properties. Firstly, they tend to have a relatively short average path length between any two nodes, even in large networks. Secondly, they exhibit high clustering due to the significant connection between nodes. Based on our research findings, NetLSD and network portrait divergence are the most suitable methods for comparing small-world topology. On the one hand, NetLSD captures connectivity between nodes by inheriting properties of Laplacian spectra. On the other hand, it verifies global and local properties, including size-invariant, permutation-invariant, and scale-adaptivity. These qualities make it ideal for analyzing large and highly connected networks, such as small-world networks. Network portrait divergence extracts node degree distribution, shortest path distribution, and next-nearest neighbors distribution, making it consistent for small-world network properties.

Therefore, global features such as eigenvalues of the Laplacian spectra, size-invariant, scale-adaptivity, degree distribution, shortest path distribution, and next-nearest neighbors distribution appear to be more effective in identifying network similarities. That is thanks to the information that global properties provide about the overall structure and characteristics of the graph, such as the connection between nodes and the features of neighboring nodes.

In this paper, we conducted a study to evaluate statistic, embedding, and Laplacian unknown node-correspondence approaches for comparing movie similarities. We found that the network portrait divergence, the NetLSD, the NetMF, the D-measure, and the Laplacian spectra performed well in determining the similarity between movies in horror, romance, sci-fi, and comedy categories. To represent movie stories as networks, we used a multilayer network model that extracts three layers from each movie script: character, keyword, location, and their interactions. We compared monolayers belonging to the same entity (characters with characters, keywords with keywords, etc.). We analyzed the similarity between movie networks by studying their structural information based on the distance between their feature vectors.

To assess the performance of measures in comparing movies, we gathered our dataset by asking people to rank the similarity between films according to their points of view. We then compared the results generated with the dataset. A measure is efficient if it produces the same results as the dataset.

According to our analysis, portrait divergence is an effective method for character layer analysis in comedy, romance, and sci-fi movies. Spectral methods, especially NetLSD, are ideal for evaluating similarity within the horror genre, particularly in character and location layers. Laplacian spectra outperformed other measures in comparing keyword layers for horror movies. NetLSD is a highly effective method for comparing movies of different genres and classifying them based on their genre. Network portrait divergence accurately categorized movies based on character and location layers, with some exceptions. Laplacian spectra excelled in comparing and classifying movies through keyword and location layers. However, NetMF and D-measure are ambiguous methods.

In general, depending on the ability of an approach property to extract network features, it can be efficient for a network type. We found that the network portrait divergence and NetLSD were effective measures for comparing character layers across various genres, and the Laplacian spectra was an effective measure for comparing keyword and location layers. Global properties are more effective than local features in capturing the connectivity between nodes, a crucial characteristic of networks. Node degree distribution, shortest-path distribution, next-nearest neighbors distribution, size-invariant, and scale-adaptivity are efficient for comparing small-world networks. On the other hand, eigenvalues of the Laplacian spectra are efficient in comparing scale-free and chain-type networks. In our future work, we will introduce an approach considering interactions between entities of the same and different entities. This approach will consider multilayers without ignoring interlayers. In other words, it will calculate the distance between multilayers based on their interrelationships and intrarelationships. Moreover, we will conduct a comparative analysis of our movie networks with a benchmark.

**Author Contributions:** Conceptualization, M.L., H.C., B.R. and M.E.H.; Methodology, M.L., H.C., B.R. and M.E.H.; Software, M.L.; Validation, H.C., B.R. and M.E.H.; Investigation, M.L.; Formal analysis, M.L., H.C., B.R. and M.E.H.; Writing—original draft, M.L., H.C., B.R. and M.E.H.; Supervision, H.C., B.R. and M.E.H. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research received no external funding.

**Data Availability Statement:** Dataset available on request from the authors.

**Acknowledgments:** We would like to thank the authors of publications for replying to our questions, especially Tiago A. Schieber, Jim Bagrow, and Youssef Mourchid. We are thankful to the reviewers for their constructive comments, which helped improve the paper.

**Conflicts of Interest:** The authors declare no conflicts of interest.

## Abbreviations

The following abbreviations are used in this manuscript:

NetLSD	Network Laplacian spectral descriptor
NetMF	Network embedding as matrix factorization
SW	Star Wars
SC	Scream

## Appendix A

### Appendix A.1. Movie Stories

**The Phantom Menace:** The story follows Master Qui-Gon-Jinn and his student Obi-One to protect queen Amidala (Padme) from the dark side. Queen Amidala lived on Naboo planet with her federation. In Tatooine, Qui-Gon discovered Anakin Skywalker, a young child in servitude with exceptional supernatural power. Darth Maul is an antagonist from the dark side, who killed Qui-Gon, and was killed by Obi-One. While dying, Qui-Gon requests Obi-Wan to train Anakin to become a Jedi.

**The Attack Of The Clones:** Ten years later, Anakin Skywalker was assigned to protect Amidala during a mission. They developed a romantic relationship and married in secret.

During the assignment, Anakin envisioned his mother in pain and decided to return to Tatooine to rescue her. Anakin Skywalker and his Master Obi-one were assigned to save the galaxy. Obi-One discovered the trick of Count Dooku, a leader from the dark side who ordered his gathering to kill Padme. Count Dooku asked Obi-One to join him, but he refused. Anakin and Padmé tried to rescue Obi-Wan from Count Dooku, but they failed and were sentenced to death. Fortunately, Masters Yoda and Mince Windu saved them from the assessment.

Revenge of the Sith: Palpatine assured Anakin to save Padme's life if he returned to the dark side. As the poor Anakin was convinced and converted to the dark side, Padme and Obi-One attempted to turn Anakin to the light side, but he refused. Moreover, Anakin strangled Padme to oblivion. Obi-One raised a lightsaber battle against Anakin on Mustafar. The fight ended with Obi-One cutting Anakin's arm and legs. Obi-One watched Anakin in horror as he was burning inside a volcano and left him for dead. Palpatine saved Anakin's life and transformed him into a cyborg (Darth Vader). Padme died giving birth to two twins: Luke and Lea.

A New Hope: Nineteen years after the Revenge of the Sith, Darth Vader imprisoned Princess Lea. Luke and Obi-One were trying to save Lea. Vader was trying to stop a rebellion, using the Death Star. Luke Skywalker and Han Solo were working to destroy the Death Star. The robots C3-PO and R2-D2 were helping the trio Lea, Luke, and Han Solo. Vader (Anakin) fought again with his Master Ben (Obi-One) and destroyed him with his lightsaber.

The Empire Strikes Back: Luke returned to Master Yoda to learn more about the Force obscure to destroy Vader. But, when Vader informed his son Luke about their relationship, Luke refused to kill his father (Vader). Moreover, Vader tried to convince Luke to return to the dark side and destroy the Emperor (Palpatine). Luke refused. Vader fought his son Luke and cut his hand. Lando, a friend of Han Solo, tracked down Luke, Han Solo, and Lea to surrender them to Palpatine. Vader froze Han-Solo. Lando freed Leia.

Return of the Jedi: The emperor constructed a new Death Star protected by a potential shield. Han Solo, Lea, and Chewbacca were searching for the shield inside a forest. Luke tried to turn Vader to the light side while Vader forced Luke to return to the dark side. In a battle between Vader and Luke, Luke removed the lightsaber from Vader and was at the point of killing him. At that moment, Palpatine encouraged Luke to kill his father and take his place, but Luke refused to destroy him, so the Emperor (Palpatine) tortured Luke with Force lightning. Vader saw his son dying. Quickly, he was thrown down in Palpatine and saved his son Luke. Unfortunately, Anakin (Vader) was dying because of electrocution. While dying, Anakin asked Luke to remove his mask to see him with his own eyes for the first time. Lando and the rebel pilot destroyed the Death Star. While Rebel fighters were celebrating their victory, Luke was looking brightly upon the Force spirits of Obi-Wan, Yoda, and his father.

Fascination: The story revolves around Bella and Edward, teens who met in a high school and developed a romantic relationship. Bella was fascinated by Edward and his family (Rene, Rosalie, Charlie). Edward and his family were vampires and had the power of reading people's ideas, except Bella. They decided to hide this secret from Bella, though Bella searched and found their secret.

New Moon: Edward and his family traveled to protect Bella's life. Bella had depression and decided to isolate herself. Jacob, Bella's friend who has the power to convert to a wolf, helped Bella out of her depression. Alice saw Bella jumping off in a vision. She informed Edward about her nightmare. Edward thought Bella was dead and decided to end his life. To do so, he went to the Volturi. Alice informed Bella about Edward's plan, so Bella went to stop Edward from dying. Bella had to be transformed into a vampire to save Edward's life.

Titanic: The story takes place on a ship called the Titanic. Rose was traveling in the first-class section of the Titanic with her family and fiance (Caledon). Jack and his friend (Fabrizio) were traveling in the third class. Rose's mother (Ruth) was forcing Rose to marry Caledon. Rose refused and decided to end her life. Jack noticed that Rose was at the point of jumping off the boat, and he saved her. Rose's family invited Jack to dinner to thank him. Rose and Jack developed a romantic relationship. Caledon accused Jack of stealing

a piece of jewelry, and he had him arrested. That same day, at night, the Titanic collided with an iceberg and was slowly flooding, so Rose looked for Jack to save him and escape. Unfortunately, when they reached the last part of the ship, the Titanic was completely drowned, so they found themselves in the cold water. Jack sacrificed his life to save Rose. Indeed, Jack agreed to die under the cold water to let Rose lie on a piece of wood.

“The Scream Saga follows a series of murders, where killers wear a ghost face.

**Scream 1 :** The first part of the story follows Casey, a young student in a Woodsboro High School. Casey was in her house when the phone rang for the first time. Casey answered the phone and thought the man’s voice (Stu) had the wrong number. The phone rang again when Casey was in the kitchen. The man’s voice asked her for her favorite scary movie. When Casey answered the question, he asked her to see her boyfriend dead in the front yard. The killer(Stu) followed Casey and killed her in the front yard. When Casey’s parents returned home, they tried to call the police but failed. Casey’s mother was screaming when she found her daughter dead. The second part follows Sidney, a teenage student in Woodsboro high school. Billy Loomis, the boyfriend of Sidney, had killed her mother with the help of Stu. That is because Sidney’s mother was the cause of the separation of Billy’s parents. One year later, Billy informed Casey about her mother’s murder and killed Stu. Sidney survived under her fight with Billy and Stu. Finally, Sidney killed Billy by shooting him with a gun.

**Scream 2:** The events of the screams reached in theatres. Luke Wilson played the role of Billy Loomis, and Tori Spelling played Sidney’s character. A series of murders began. Sidney survived again by confronting the new Ghostface killers. With the help of Mickey(Sidney’s friend), Mrs. Loomis(Billy’s mother) tried to avenge her son’s died. Mickey killed Derek, the new boyfriend of Sidney. Mrs. Loomis killed Randy(Sidney’s friend) because he bad-mounted Billy. Then, she killed Mickey because she found him useless. Mrs. Loomis confronted Sidney. Mrs. Loomis was killed by Cotton Weary, who was accused of murdering Sidney’s mother.

**Scream 3:** Sidney went to the mountain to hide from the killers. She received a call from the Ghostface(Roman). First, she was thinking the caller’s voice was of her mother. Roman, Ghostface’s new killer, had executed a series of murders. He killed Cotton, Steven, and others. In the end, he was killed by Sidney and Dewey.” [28]

**500 Days of Summer:** The movie follows the relationship between Summer and Tom during 500 days. On day 1, Summer started working in the same company as Tom. On day 28, Tom fell in love with Summer and believed she was the right person, but Summer refused to engage in a relationship. After some months, they had a fake relationship. Summer decided to break it off on day 290 and quit her job to let Tom live his life. On day 476, Summer got married to Seth. Tom felt depressed, but he wished her happiness. On day 500, Tom met another woman named Automne.

**Ten Things I Hate About You:** The story follows the sisters Bianca and Kat. Their father was controlling their lives. Kat, the older sister of Bianca, has been accepted into a school far from home. But her father refuses to let her go to the school, to keep her close to home. Kat failed in making relationships with school friends because of her antisocial personality. The father prevented Bianca from dating her boyfriend (Cameron) until her older sister got a boyfriend. Cameron paid Joey to date Kat so that her father would allow Bianca to date him.

**Airplane:** The story follows Striker, an ex-fighter pilot, and a taxi driver. Elaine, his girlfriend during the war, became a flight attendant and broke up with him. Despite his flying phobia, Striker bought a ticket and flew on the same airplane as Elaine. Striker had the intention to get Elaine back, but she refused. All the passengers were crying because of fish poisoning. Elaine called a supervisor (McCroskey) to activate the autopilot. As the airplane pilot failed in controlling the plane, Elaine and Dr. Rumack persuaded Striker to drive the airplane. McCroskey called Kramer to help Striker in control, but Striker was

uncomfortable with McCroskey’s orders and lost control. Fortunately, Dickinson and Dr. Rumack encouraged Striker to maintain airplane control again. Despite the weather being worse near Chicago, Striker landed the airplane safely. Elaine was impressed by Striker’s courage and went back to him.

Appendix A.2. Movie Networks Visualization

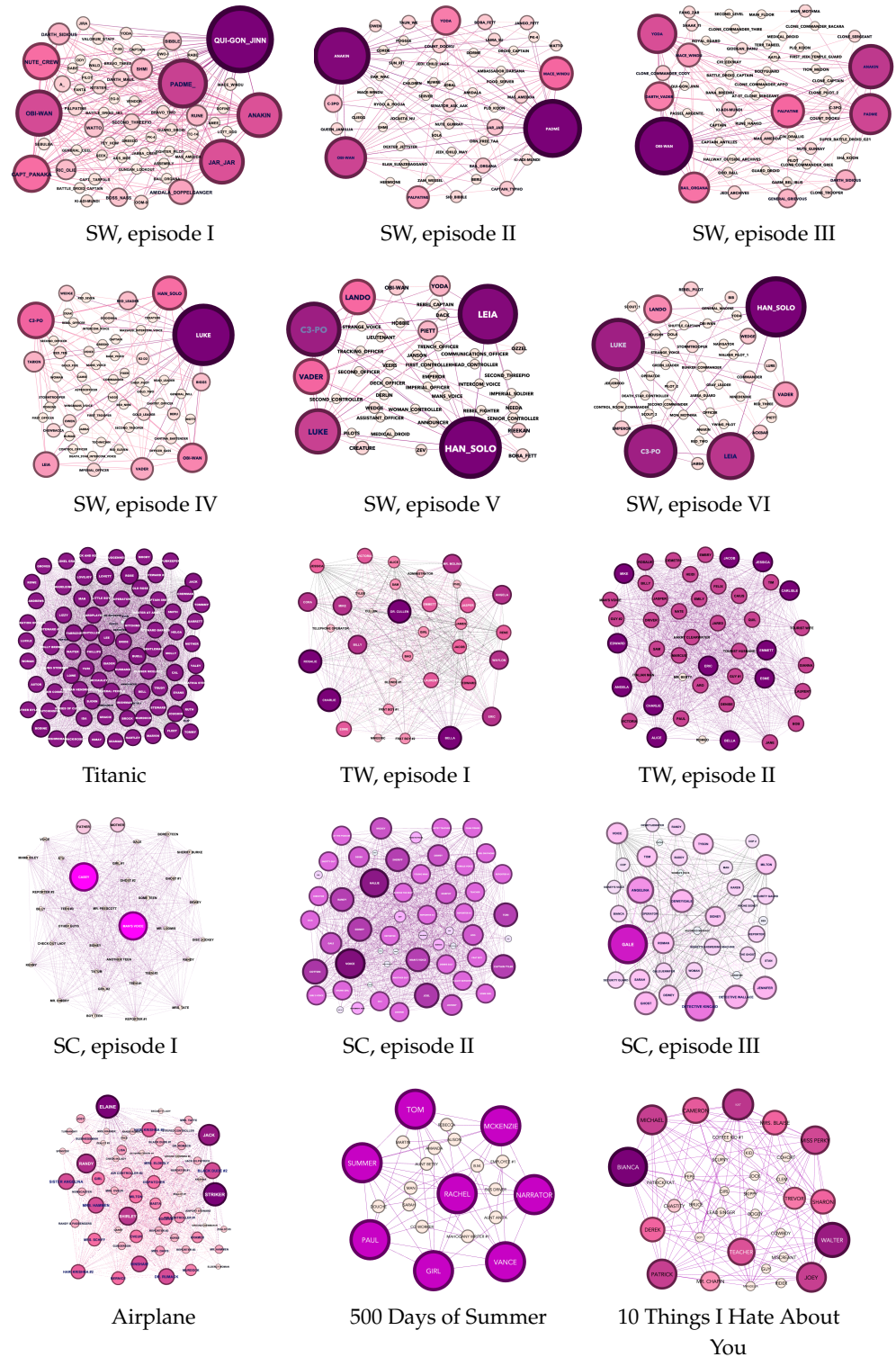


Figure A1. Visualization of character networks in sci-fi, romance, horror, and comedy movies.

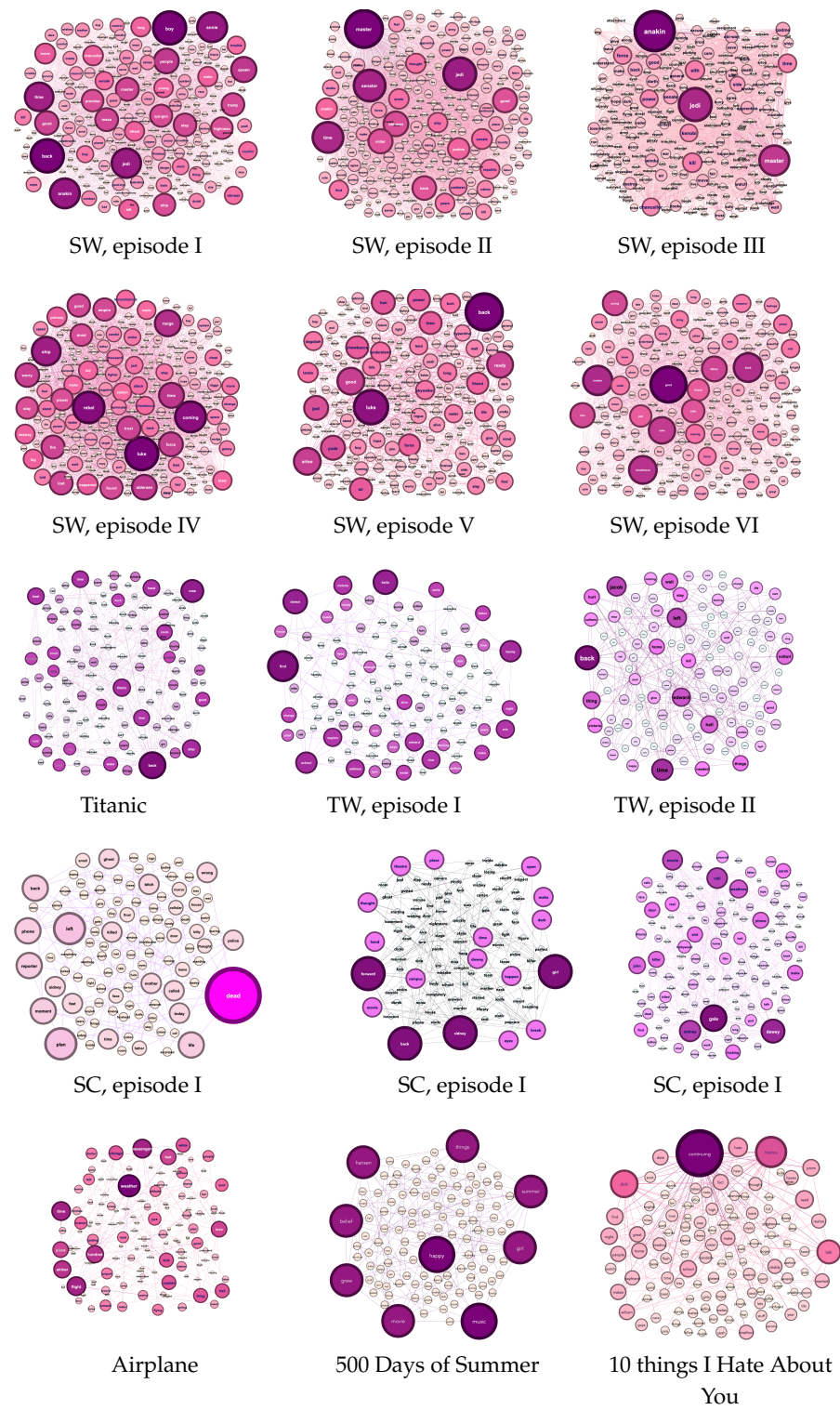


Figure A2. Visualization of keyword networks in sci-fi, romance, horror, and comedy movies.

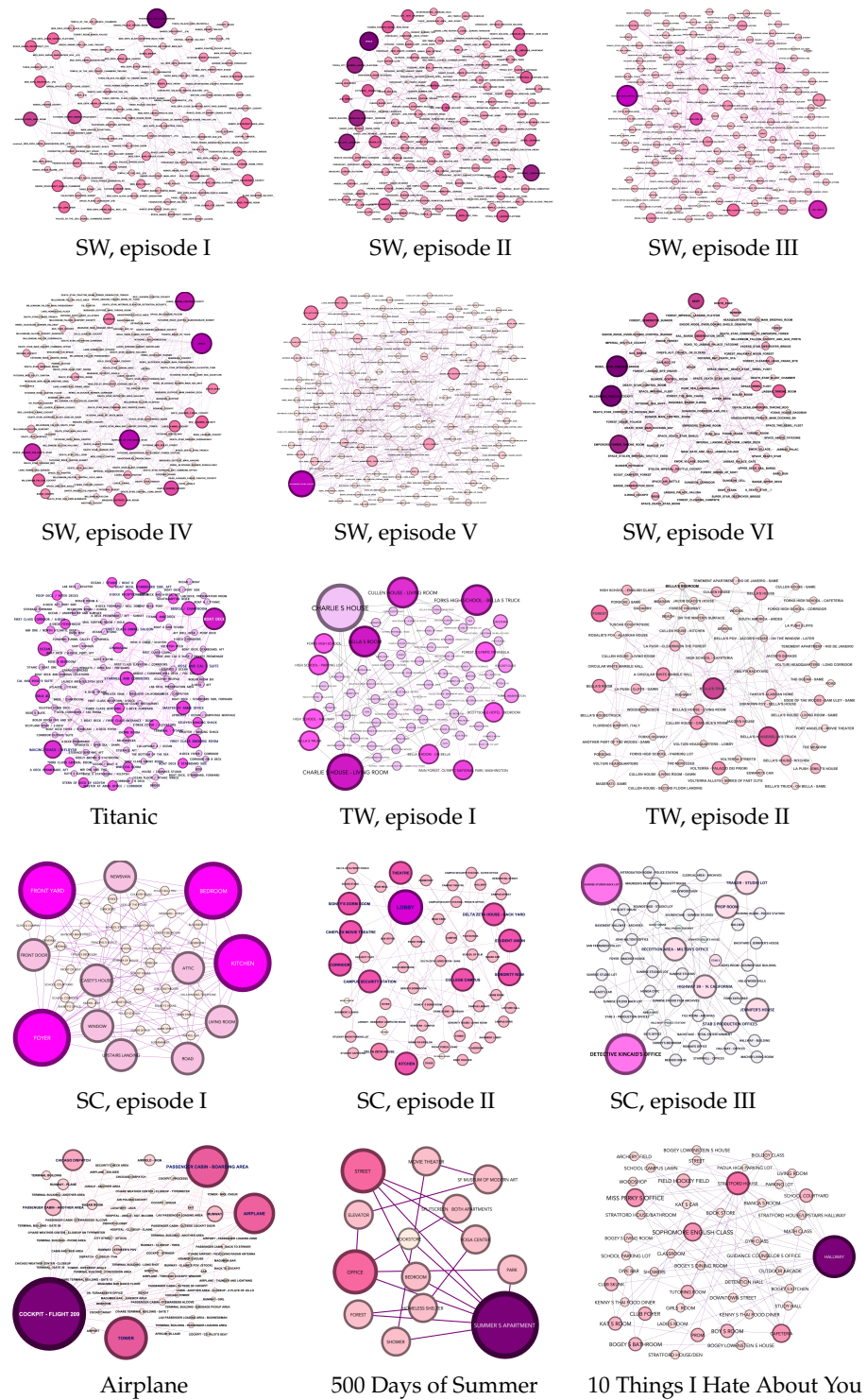


Figure A3. Visualization of location networks in sci-fi, romance, horror, and comedy movies.

References

1. Kardan, A.A.; Ebrahimi, M. A novel approach to hybrid recommendation systems based on association rules mining for content recommendation in asynchronous discussion groups. *Inf. Sci.* **2013**, *219*, 93–110. [CrossRef]
2. Drif, A.; Zerrad, H.E.; Cherifi, H. Ensvae: Ensemble variational autoencoders for recommendations. *IEEE Access* **2020**, *8*, 188335–188351. [CrossRef]

3. Drif, A.; Guembour, S.; Cherifi, H. A sentiment enhanced deep collaborative filtering recommender system. In Proceedings of the Complex Networks & Their Applications IX: Volume 2, Proceedings of the Ninth International Conference on Complex Networks and Their Applications COMPLEX NETWORKS 2020, Madrid, Spain, 1–3 December 2020; Springer International Publishing: Berlin/Heidelberg, Germany, 2021; pp. 66–78.
4. Drif, A.; Cherifi, H. Migan: Mutual-interaction graph attention network for collaborative filtering. *Entropy* **2022**, *24*, 1084. [[CrossRef](#)]
5. Sang, J.; Xu, C. Character-based movie summarization. In Proceedings of the 18th ACM international conference on Multimedia, Firenze, Italy, 25–29 October 2010; pp. 855–858.
6. Tran, Q.D.; Hwang, D.; Lee, O.J.; Jung, J.E. Exploiting character networks for movie summarization. *Multimed. Tools Appl.* **2017**, *76*, 10357–10369. [[CrossRef](#)]
7. Li, Y.; Narayanan, S.; Kuo, C.C.J. Content-based movie analysis and indexing based on audiovisual cues. *IEEE Trans. Circuits Syst. Video Technol.* **2004**, *14*, 1073–1085. [[CrossRef](#)]
8. Adams, B.; Dorai, C.; Venkatesh, S. Toward automatic extraction of expressive elements from motion pictures: Tempo. *IEEE Trans. Multimed.* **2002**, *4*, 472–481. [[CrossRef](#)]
9. Weng, C.Y.; Chu, W.T.; Wu, J.L. Rolenet: Movie analysis from the perspective of social networks. *IEEE Trans. Multimed.* **2009**, *11*, 256–271. [[CrossRef](#)]
10. Jung, J.J.; You, E.; Park, S.B. Emotion-based character clustering for managing story-based contents: A cinemetric analysis. *Multimed. Tools Appl.* **2013**, *65*, 29–45. [[CrossRef](#)]
11. Weng, C.Y.; Chu, W.T.; Wu, J.L. Movie analysis based on roles' social network. In Proceedings of the 2007 IEEE International Conference on Multimedia and Expo, Beijing, China, 2–5 July 2007; IEEE: Piscataway, NJ, USA, 2007; pp. 1403–1406.
12. Mouchid, Y.; Renoust, B.; Cherifi, H.; El Hassouni, M. Multilayer network model of movie script. In Proceedings of the Complex Networks and Their Applications VII: Volume 1 Proceedings The 7th International Conference on Complex Networks and Their Applications COMPLEX NETWORKS, Cambridge, UK, 11–13 December 2018; Springer: Berlin/Heidelberg, Germany, 2019; pp. 782–796.
13. Markovič, R.; Gosak, M.; Perc, M.; Marhl, M.; Grubelnik, V. Applying network theory to fables: Complexity in Slovene belles-lettres for different age groups. *J. Complex Netw.* **2019**, *7*, 114–127. [[CrossRef](#)]
14. Lv, J.; Wu, B.; Zhou, L.; Wang, H. Storyrolenet: Social network construction of role relationship in video. *IEEE Access* **2018**, *6*, 25958–25969. [[CrossRef](#)]
15. Chen, R.G.; Chen, C.C.; Chen, C.M. Unsupervised cluster analyses of character networks in fiction: Community structure and centrality. *Knowl.-Based Syst.* **2019**, *163*, 800–810. [[CrossRef](#)]
16. Mouchid, Y.; Renoust, B.; Roupin, O.; Vãn, L.; Cherifi, H.; Hassouni, M.E. Movienet: A movie multilayer network model using visual and textual semantic cues. *Appl. Netw. Sci.* **2019**, *4*, 1–37. [[CrossRef](#)]
17. Xiao, Q. A method for measuring node importance in hypernetwork model. *Res. J. Appl. Sci. Eng. Technol.* **2013**, *5*, 568–573. [[CrossRef](#)]
18. Das, K.; Samanta, S.; Pal, M. Study on centrality measures in social networks: A survey. *Soc. Netw. Anal. Min.* **2018**, *8*, 1–11. [[CrossRef](#)]
19. Abdelsadek, Y.; Chelghoum, K.; Herrmann, F.; Kacem, I.; Otjacques, B. Community extraction and visualization in social networks applied to Twitter. *Inf. Sci.* **2018**, *424*, 204–223. [[CrossRef](#)]
20. Grandjean, M. Comparing the Relational Structure of the Gospels. Network Analysis as a Tool for Biblical Sciences. Society of Biblical Literature. 2013. Available online: <https://hal.science/hal-01525574/file/Grandjean-2013.pdf> (accessed on 1 December 2023).
21. Bastian, M.; Heymann, S.; Jacomy, M. Gephi: An open source software for exploring and manipulating networks. In Proceedings of the Third international AAAI Conference on Weblogs and Social Media, San Jose, CA, USA, 17–20 May 2009.
22. Schieber, T.A.; Carpi, L.; Díaz-Guilera, A.; Pardalos, P.M.; Masoller, C.; Ravetti, M.G. Quantification of network structural dissimilarities. *Nat. Commun.* **2017**, *8*, 1–10. [[CrossRef](#)] [[PubMed](#)]
23. Zhang, R.J.; Fred, Y.Y. Measuring similarity for clarifying layer difference in multiplex ad hoc duplex information networks. *J. Inf.* **2020**, *14*, 100987. [[CrossRef](#)]
24. Saxena, R.; Kaur, S.; Bhatnagar, V. Identifying similar networks using structural hierarchy. *Phys. A Stat. Mech. Appl.* **2019**, *536*, 121029. [[CrossRef](#)]
25. Bródka, P.; Chmiel, A.; Magnani, M.; Ragozini, G. Quantifying layer similarity in multiplex networks: A systematic study. *R. Soc. Open Sci.* **2018**, *5*, 171747. [[CrossRef](#)]
26. Lafhel, M.; Cherifi, H.; Renoust, B.; El Hassouni, M.; Mouchid, Y. Movie Script Similarity Using Multilayer Network Portrait Divergence. In Proceedings of the International Conference on Complex Networks and Their Applications, Madrid, Spain, 1–3 December 2020; Springer: Berlin/Heidelberg, Germany, 2020; pp. 284–295.
27. Bagrow, J.P.; Bollt, E.M.; Skufca, J.D.; Ben-Avraham, D. Portraits of complex networks. *EPL (Europhys. Lett.)* **2008**, *81*, 68004. [[CrossRef](#)]
28. Lafhel, M.; Abrouk, L.; Cherifi, H.; El Hassouni, M. The similarity between movie scripts using Multilayer Network Laplacian Spectra Descriptor. In Proceedings of the 2022 IEEE Workshop on Complexity in Engineering (COMPENG), Florence, Italy, 18–20 July 2022; IEEE: Piscataway, NJ, USA, 2022; pp. 1–5.
29. Blei, D.M.; Ng, A.Y.; Jordan, M.I. Latent dirichlet allocation. *J. Mach. Learn. Res.* **2003**, *3*, 993–1022.



30. Wilson, R.C.; Zhu, P. A study of graph spectra for comparing graphs and trees. *Pattern Recognit.* **2008**, *41*, 2833–2841. [[CrossRef](#)]
31. Zhu, P.; Wilson, R.C. A study of graph spectra for comparing graphs. In Proceedings of the BMVC, Oxford, UK, 5–8 September 2005.
32. Tsitsulin, A.; Mottin, D.; Karras, P.; Bronstein, A.; Müller, E. Netlsd: Hearing the shape of a graph. In Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, London, UK, 19–23 August 2018; pp. 2347–2356.
33. Goyal, P.; Ferrara, E. Graph embedding techniques, applications, and performance: A survey. *Knowl.-Based Syst.* **2018**, *151*, 78–94. [[CrossRef](#)]
34. Cui, P.; Wang, X.; Pei, J.; Zhu, W. A survey on network embedding. *IEEE Trans. Knowl. Data Eng.* **2018**, *31*, 833–852. [[CrossRef](#)]
35. Cai, H.; Zheng, V.W.; Chang, K.C.C. A comprehensive survey of graph embedding: Problems, techniques, and applications. *IEEE Trans. Knowl. Data Eng.* **2018**, *30*, 1616–1637. [[CrossRef](#)]
36. Li, B.; Pi, D. Learning deep neural networks for node classification. *Expert Syst. Appl.* **2019**, *137*, 324–334. [[CrossRef](#)]
37. Liao, L.; He, X.; Zhang, H.; Chua, T.S. Attributed social network embedding. *IEEE Trans. Knowl. Data Eng.* **2018**, *30*, 2257–2270. [[CrossRef](#)]
38. Wang, C.; Pan, S.; Hu, R.; Long, G.; Jiang, J.; Zhang, C. Attributed graph clustering: A deep attentional embedding approach. *arXiv* **2019**, arXiv:1906.06532.
39. Ding, C.H.; He, X.; Zha, H.; Gu, M.; Simon, H.D. A min-max cut algorithm for graph partitioning and data clustering. In Proceedings of the 2001 IEEE International Conference on Data Mining, San Jose, CA, USA, 29 November–2 December 2001; IEEE: Piscataway, NJ, USA, 2001; pp. 107–114.
40. Chen, H.; Yu, Z.; Yang, Q.; Shao, J. Attributed graph clustering with subspace stochastic block model. *Inf. Sci.* **2020**, *535*, 130–141. [[CrossRef](#)]
41. Liu, F.; Xue, S.; Wu, J.; Zhou, C.; Hu, W.; Paris, C.; Nepal, S.; Yang, J.; Yu, P.S. Deep learning for community detection: Progress, challenges and opportunities. *arXiv* **2020**, arXiv:2005.08225.
42. Van der Maaten, L.; Hinton, G. Visualizing data using t-SNE. *J. Mach. Learn. Res.* **2008**, *9*, 2579–2605.
43. Chen, X.; Heimann, M.; Vahedian, F.; Koutra, D. CONE-Align: Consistent Network Alignment with Proximity-Preserving Node Embedding. In Proceedings of the 29th ACM International Conference on Information & Knowledge Management, Virtual Event, 19–23 October 2020; pp. 1985–1988.
44. Asta, D.; Shalizi, C.R. Geometric network comparison. *arXiv* **2014**, arXiv:1411.1350.
45. Huang, W.; Ribeiro, A. Network comparison: Embeddings and interiors. *IEEE Trans. Signal Process.* **2017**, *66*, 412–427. [[CrossRef](#)]
46. Koren, Y.; Bell, R.; Volinsky, C. Matrix factorization techniques for recommender systems. *Computer* **2009**, *42*, 30–37. [[CrossRef](#)]
47. Van Loan, C.F. Generalizing the singular value decomposition. *SIAM J. Numer. Anal.* **1976**, *13*, 76–83. [[CrossRef](#)]
48. Perozzi, B.; Al-Rfou, R.; Skiena, S. Deepwalk: Online learning of social representations. In Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, New York, NY, USA, 24–27 August 2014; pp. 701–710.
49. Qiu, J.; Dong, Y.; Ma, H.; Li, J.; Wang, K.; Tang, J. Network embedding as matrix factorization: Unifying deepwalk, line, pte, and node2vec. In Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining, Marina Del Rey, CA, USA, 5–9 February 2018; pp. 459–467.
50. Bagrow, J.P.; Boltt, E.M. An information-theoretic, all-scales approach to comparing networks. *Appl. Netw. Sci.* **2019**, *4*, 45. [[CrossRef](#)]

**Disclaimer/Publisher’s Note:** The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.