MoGaL: Novel Movie Graph Construction by Applying LDA on Subtitle
Mohammad Nazari, Hossein Rahmani*, Dadfar Momeni and Motahareh Nasiri
School of Computer Engineering, Iran University of Science and Technology, Tehran, Iran.

Abstract
Graph representation of data can better define the relationships among the data components, and thus provide a better and richer analysis. So far, movies have been represented in graphs many times using different features for clustering, genre prediction, and even for use in recommender systems. In constructing movie graphs, little attention has been paid to their textual features such as subtitles, while they contain the entire content of the movie, and there is a lot of hidden information in them. Thus in this paper, we propose a method called MoGaL to construct movie graph using LDA on subtitles. In this method, each node is a movie, and each edge represents the novel relationship discovered by MoGaL among two associated movies. First, we extract the important topics of the movies using LDA on their subtitles. Then we visualize the relationship between the movies in a graph using the cosine similarity. Finally, we evaluate the proposed method with respect to the measures genre homophily and genre entropy. MoGaL succeeds to outperform the baseline method significantly in these measures. Accordingly, our empirical results indicate that movie subtitles could be considered a rich source of informative information for various movie analysis tasks.

1. Introduction
Movies are a big part of the entertainment industry, and their number is increasing daily. With the spread of online streaming websites and people's acceptance, this industry has become more important, so this data has attracted the attention of the data scientists and researchers in the recent years [1-5].

One of the most useful and widely used approaches in the field of data mining and relationship discovery is the representation of the problem in the form of a graph. Graph representation can be very useful in visualizing and understandably presenting data [6, 7]. Movie graphs allow for studying relationships, discovering similarities between movies, and using graph analysis algorithms. In various papers, movie graphs have been used to recommend movies to the users [8-10], genre predictions [11], find cooperation patterns between movie actors, and suggest actors to directors [4, 12]. It has been shown that using more data types enriches the graph, and leads to better results [13-15]. In order to construct graph, they usually used the relationships among features that describe the movie such as writer, director, and actors or user ratings to the movies [4, 12, 9].

To the best of our knowledge, there is no previous work that used the textual data of movies in the construction of movie graph, and this important feature of movies has been neglected. The textual features of the movies indicate the content of the movie. Subtitles, plot synopsis, and summary can be mentioned as the textual features of the movies. These unstructured features contain a lot of information that can be used to represent the content of a movie [1, 10].

According to the recent advances in the field of natural language processing, various analyzes have been performed to find the content similarity of movies using textual features, which include all
kinds of traditional methods, machine learning, and deep learning [1-3, 10]. Therefore, we want to obtain the content relationships between movies using subtitles and represent it in a graph. We enriched movie graphs and we succeeded to get better results by adding content relations of movies to them. In this paper, we propose a method called MoGaL (Movie Graph Construction by Applying LDA) on subtitle that uses subtitles similarity to construct a movies graph. This method first extracts the important topics of movies using LDA [16] on their subtitles. Then constructs a movie graph using the topic similarities. Finally, considering that the genre is a stylistic or thematic categorization based on the main story of the movie, we evaluate the graph made with three methods based on the genre of the movies.

The structure of this paper is as what follows. Section 2 overviews the available methods for constructing the movie graphs and measuring the similarity of movies using textual features. The proposed method to construct a movies graph is described in detail in Section 3. The Empirical results are presented in Section 4. Section 5 includes a discussion of the results, and proposes promising directions for future research works.

2. Background

Graph representation of data makes it easier to understand and analyze them; this is also true for movies [6, 7]. Movies have frequently been modeled as graph in papers for various purposes [12, 17, 18]. One of the most popular methods for representing movies in graphs is to consider each movie as a node connected by links to features that describe the movie such as writer, director, actors, studio, year of production, and awards [7]. Toine [19] used this graph to build a contextual recommendation model. Toine was able to increase the accuracy of his movie recommender system by using the contextual information of the movies that he put in the graph. Also by analyzing this graphs, Tang et al. [4] found the relationships between the types of movies and the cooperation networks of the directors and the actors, and Spitz et al. [18] predicted the probability of success and popularity of a movie. Another common way to construct movie graph is making a bipartite graph of movies and users, where each edge represents the user's score for each movie. This graph is widely used in recommender systems [8, 9]. In this graph, by using different machine learning methods, it is possible to predict the score of users, and suggest movies to the users. Lee et al. [14] added new edges to the bipartite graph that represent users' emotions toward movies to have a better recommender system. Also, in addition to the bipartite graph of users' scores, Darban et al. [20] used side-information of the users to recommend movies. Zhou et al. [11] also used the graph of movies to predict the genre. In their graph, each movie is a node, and each edge represents having the same director. The tags assigned to each movie on the IMDb website are also considered attributes of each movie in the graph.

As we have reviewed, the movie graph has many applications, and each of them has used different features of the movies to construct the graph. However, we could not find any paper that uses textual data in graph construction. Therefore, the content relationship between movies on these networks has been neglected. The textual features of the movies indicate the content of the movie. Subtitles, plot synopsis, and summary can be mentioned as the textual features of the movies. These unstructured features contain a lot of information that can be very useful if extracted correctly [13]. Due to the advances in natural language processing, subtitles have been used in many papers for various applications. For example, Bougiatiotis et al. [2] represented each movie directly using its subtitles, and used it to calculate the similarity between the movies [21]. Luhmann et al. [1] introduced the SubRosa method, which used subtitles to solve the cold start [15] problem in the recommender systems. SubRosa is a content-based recommender system based on subtitles of movies. Also Hasan et al. [3] used movie subtitles to cluster the movies. We have seen those subtitles, as the longest textual feature of movies, have already been used in the representation and analysis of movies but they have never been used to construct movies graph. Therefore, in this work, we want to construct a graph of movies using their subtitles.

3. Proposed Method

In this section, we propose a method called MoGaL for constructing the graph of the movies based on LDA on their subtitles, and then we evaluate the constructed graph based on three measures. In this method, after collecting the data and pre-processing it, we construct the graph of the movies using the proposed method MoGaL. In the following, we describe MoGaL in more detail. After constructing the graph, we evaluate it using 3 proposed measures based on genre homophily and genre entropy. Figure 1 shows the structure of the proposed method, each of its steps will be explained in detail in the next sections.
3.1. Dataset
Our dataset includes the structured and unstructured features. Structured data refers to movie features such as title, release year, and genres, which we used the IMDb online dataset [22] to collect. Unstructured data refers to the English subtitles of the movies, which we collected using website crawling. We represent our dataset with $M$, which contains structured and unstructured features for 4050 movies. Table 1 describes our dataset.

3.2. Pre-processing
Subtitles include daily conversation dialogues and special characters that we clean and ready to analyze in several steps.
- Removing HTML tags that include information about the subtitle-provider website, movie specifications, and details about how subtitles are displayed including font type and color.
- Removing additional descriptions that describe some of the important sounds of the film for the hearing-impaired such as hums, whispers, cheers, and passing cars.

Table 1. Description of each database column.

<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>$M$</td>
<td>Movies set</td>
<td>$M = { m_1, m_2, \ldots, m_{4050} }, \quad</td>
</tr>
<tr>
<td>GENRES</td>
<td>Genres set</td>
<td>${ \text{Action}, \text{Adventure}, \text{Animation}, \text{Biography}, \text{Comedy}, \text{Crime}, \text{ocumentary}, \text{Drama}, \text{Family}, \text{Fantasy}, \text{FilmNoir}, \text{History}, \text{Horror}, \text{Music}, \text{Musical}, \text{Mystery}, \text{News}, \text{Romance}, \text{Sci-Fi}, \text{Sport}, \text{Thriller}, \text{War}, \text{Western} }$, $</td>
</tr>
<tr>
<td>$Gen_{m_i}$</td>
<td>Genes set for movie $m_i$</td>
<td>Each movie $m_i$ in our dataset can have 1 to 3 genres, which we denote by $Gen_{m_i}$, $Gen_{m_i} = { g_{m_i1}, \ldots, g_{m_in} }, \quad 1 \leq n \leq 3, \quad g \in \text{GENRES}$</td>
</tr>
<tr>
<td>$GenVec_{m_i}$</td>
<td>Genre vector for movie $m_i$</td>
<td>$GenVec_{m_i} = { b_1, b_2, b_3, \ldots, b_n }, \quad n =</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$b \in (0,1)$, $b_i = 1$ if $g_i \in Gen_{m_i}$, $b_i = 0$ if $g_i \notin Gen_{m_i}$</td>
</tr>
</tbody>
</table>
• Removing frequently used words such as plural signs and prepositions, which do not have an influential connotation and can be ignored (stop words).
• Removing specific names that are repeated in a large number in a subtitle, and do not have an influential conceptual load using the NER tool and the spacy library [23].
• Tokenizing a subtitle into a list of words, and lemmatizing each word.

3.3. Movie Graph Construction: MoGaL

In this section, we construct the graph of movies using the MoGaL algorithm. First, we use the LDA algorithm to extract important topics from the subtitles. In order to determine the number of topics of LDA, we calculate the coherence and perplexity for the different number of topics and denote it by t. Then we represent each movie \( m_i \) with a vector \( v_{m_i} \) of length t by applying the LDA algorithm on their subtitles. Next, to calculate the similarity between each pair of movies \( m_i \) and \( m_j \), we use the cosine similarity between vectors \( v_{m_i} \) and \( v_{m_j} \) according to (1). In this way, we calculate the cosine similarity between the movie vector two by two. Then we use these values to build the similarity matrix of the movies.

\[
\text{Cosine Similarity}(m_i, m_j) = S_c(v_{m_i}, v_{m_j}) \tag{1}
\]

Using the similarity matrix generated from the previous step, we create the similarity graph of the movies \( G(V, E) \). Graph \( G \) is a complete graph that each vertex \( v_i \in V \) represents a movie \( m_i \), and each edge \( e_{ij} \in E \) has a weight \( w_{ij} \), which shows the cosine similarity between two vertices connected to that edge, which is defined as (2).

\[
e = \forall v_i, v_j \in V \rightarrow \exists e_{ij}, w_{ij} = S_c(v_{m_i}, v_{m_j}) \tag{2}
\]

Considering all the edges and vertices in the graph, analyzing and processing the graph is more laborious and time-consuming, and therefore, the graph should be pruned. We prune the graph based on the weight of its edges. For this purpose, we choose a threshold \( \theta \) for minimum edge weight (minimum similarity), and we call the pruned graph \( G_M(V_M, E_M) \). Graph \( G_M \) is the final output of the MoGaL method.

3.4. Graph Evaluation

In this section, we evaluate the graph constructed by MoGaL. We use three different measures based on movie genre to measure the quality of the constructed graph. Genre is a stylistic or thematic categorization based on the main story of the movie, suggested by humans for each movie. Genres are widely used to categorize and recommend movies to users. Thus we can measure the effectiveness of the graph constructed by MoGaL in tasks such as recommending movies, predicting genres, and classification movies using genre.

3.4.1 Homophily

To measure the homophily in this graph, we use the Jaccard similarity measure between the set of genres of the connected movies. Thus we calculate the homophily \( (H) \) between two connected movies \( m_i \) and \( m_j \) using (3).

\[
H(m_i, m_j) = \frac{\text{Jaccard}(\text{Gen}_{m_i}, \text{Gen}_{m_j})}{\text{Gen}_{m_i} \cup \text{Gen}_{m_j}} \tag{3}
\]

To measure homophily graph \( G_M \), we use the average homophily of all connected movies \( m_i \) and \( m_j \) in graph \( G_M (e_{m_i, m_j} \in G_M) \), according to (4).

\[
H(G_M) = \frac{\sum_{e_{m_i, m_j} \in G_M} H(m_i, m_j)}{E_M} \tag{4}
\]

3.4.2 Entropy

The entropy measure is another method for evaluating the \( G_M \). In this regard, we define the entropy of each movie \( m_i \) in \( G_M \) to be equal to genre entropy of its neighbors in the graph. First, we define the set of neighbors of the movie \( m_i \) according to (5), and call it \( N_{m_i} \):

\[
N_{m_i} = \{ m_j : e_{m_i, m_j} \in E_M \} \tag{5}
\]

Then we add up the GenVec of the \( m_i \)’s neighbors (GenVec\_{N_{m_i}}) according to (6).

\[
\text{GenVec}_{N_{m_i}} = \sum_{v_j \in N_{m_i}} \text{GenVec}_{v_j} \tag{6}
\]

Finally, the entropy for each movie \( m_i \) is calculated by (7), where \( P_i \) is defined by (8). In (8), GenVec\^{i}_{N_{m_i}} represents the \( j \)-th element of the vector GenVec\_{N_{m_i}}:

\[
\text{Entropy}(m_i) = \sum_{i=1}^{\text{GENRES}} -P_i \log(P_i) \tag{7}
\]

\[
P_j = \frac{\text{GenVec}^j_{N_{m_i}}}{\sum_{k=0}^{\text{GENRES}-23} \text{GenVec}^k_{N_{m_i}}} \tag{8}
\]
Next, after calculating the entropy for each of the movies of the graph $G_M$, we calculate the entropy of the $G_M$ using the average entropy of all movies according to (9).

$$\text{Entropy}(G_M) = \frac{\sum_{v_i \in V_M} \text{Entropy}(m_i)}{V_M} \quad (9)$$

### 3.4.3. Clustering

We use clustering and entropy as another method to evaluate the graph $G_M$. For this purpose, we embed each node in the graph with the Node2Vec method [24] for the input of the clustering algorithm. This method works similar to the Word2Vec [25] method, which is common in text analysis. Then we partition the graph into $k$ clusters $c_1$ to $c_k$ using the K-means algorithm. We used WSCC (the sum of squared distance between each point and the centroid in a cluster) to find the optimal number of clusters. Then for each cluster $c_i$, we define the entropy of its members with respect to their genre vectors as follows. First, we calculate genre vector cluster $\text{GenVec}_{c_i}$ by using the sum of the GenVec of all the movies in that cluster according to (10). Then we calculate the entropy for each cluster as the same (7) and (8). After calculating the entropy of each cluster, we calculate the clusters average entropy of graph $G_M$.

$$\text{GenVec}_{c_i} = \sum_{v_i \in c_i} \text{GenVec}_{v_i} \quad (10)$$

### 4. Empirical Results

In this section, we implement MoGaL, and examine its results. The results of each evaluation measures are mentioned in the following section. We use the correlation criterion to determine the number of LDA topics. As shown in Table 2, the correlation is maximized at the value of 150. Thus the value of $t = 150$ is considered for the number of LDA topics. Then after feature extraction using LDA, we construct the graph.

<table>
<thead>
<tr>
<th>Number of topics</th>
<th>Coherence</th>
</tr>
</thead>
<tbody>
<tr>
<td>50</td>
<td>0.4554</td>
</tr>
<tr>
<td>100</td>
<td>0.5195</td>
</tr>
<tr>
<td>150</td>
<td>0.5446</td>
</tr>
<tr>
<td>200</td>
<td>0.5267</td>
</tr>
</tbody>
</table>

After constructing the complete graph of movies, to prune it, we keep only highly weighted edges that have weight more than $\theta$ ($\text{weight} \left( e_{i,j} \right) > \theta$). Figure 2 and Figure 3 show the number of nodes and edges, respectively, with respect to threshold $\theta$ in pruned graph $G_M(V_M, E_M)$. As Figure 2 and Figure 3 indicate, $\theta = 0.9$ is the best optimal value for this hyperparameter. Accordingly, the number of edges of the pruned graph $G_M(V_M, E_M)$ is reduced to 34512, and the number of its vertices with edges is reduced to 2933. The diameter of $G_M$ graph is 25, and the average of its shortest paths is 6.72. Also according to the importance of genres in our evaluation measures, The genres distribution diagram of $G_M$’s movies is given in Figure 4.
In this graph, $G_E$, which is constructed using common entities of movies. Zhou et al. [11] used this graph to classification movies by genre. In this graph, each movie is a node and each edge indicates the existence of the same director, writer or star\(^1\) between the two movies. Using this idea, we construct the graph $G_E$ for 2933 movie which exist in the graph $G_M$. Finally, the $G_E$ included 31843 edges, that 22198 of which are also exist in $G_M$. The second graph is the random graph $G_{NS}$, which we use as a baseline for comparison. We create a random graph $G_{NS}(V_{NS}, E_{NS})$ with the negative sampling method. For this purpose, we select all the nodes of the graph $G_M$. So, $V_{NS} = V_M$. Then we select the set of edges of the graph $G_{NS}$ in the following ways: first, not be a member of the set of edges of the $G_M$ ($e_{i,j} \notin E_M$). Second, they are randomly selected. And third, the number of edges of the random graph is equal to the number of edges of the $G_M$, Therefore $|E_{NS}| = |E_M|$.

Now, we measure the quality of the MoGaL for constructing movies graph by comparing the three evaluation measures discussed in the three constructed graphs. As it is shown in Table 3, the homophily in the graph constructed by MoGaL is more than the others, which indicates that the movies connected by edges have more genre similarities. The difference between the graph constructed by MoGaL and the random graph is significantly large. The values of the standard deviation in this table indicates the dispersion of the Jaccard similarity, which is good if the value is low, but on the condition that the average value is high. For example, in the random graph, a low value of the standard deviation indicates that the Jaccard similarity values are concentrated around the mean, which is a low value.

**Table 3. Genre homophily in the graphs.**

<table>
<thead>
<tr>
<th></th>
<th>$G_{NS}$</th>
<th>$G_{En}$</th>
<th>$G_M$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mean Jaccard similarity ($\mu$)</td>
<td>0.1775</td>
<td>0.3761</td>
<td>0.3957</td>
</tr>
<tr>
<td>Jaccard similarity STD ($\sigma$)</td>
<td>0.2207</td>
<td>0.2755</td>
<td>0.2973</td>
</tr>
</tbody>
</table>

In Table 4, we present the results of the genre entropy analysis of the neighbors of a node in three constructed graphs.

**Table 4. Genre entropy in the graphs.**

<table>
<thead>
<tr>
<th></th>
<th>$G_{NS}$</th>
<th>$G_{En}$</th>
<th>$G_M$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mean entropy ($\mu$)</td>
<td>3.4509</td>
<td>2.2585</td>
<td>2.2339</td>
</tr>
<tr>
<td>Entropy STD ($\sigma$)</td>
<td>0.1693</td>
<td>0.5620</td>
<td>0.6034</td>
</tr>
</tbody>
</table>

\(^1\) A movie star is an actor or actress who is famous for their starring, or leading, roles in movies. In our dataset, the maximum number of stars for each movie is 3.

A high entropy means a uniform distribution of neighbors’ genre. Thus in this case, low entropy is good. A low entropy indicates that the neighbors of a node have the same genres. As you can see in Table 4, the lowest amount of genre entropy belongs to graph $G_M$. As a result, it can be stated that the $G_M$ graph has much more homogeneity in terms of genre.

As regards the last assessment method, after performing the clustering, we calculate the clusters' average entropy for $G_M$. We use within cluster sum of squares (WSSC) to find the optimal number of clusters. Finally, we set the number of clusters to $k = 40$ according to Figure 5. Table 5 shows the average and standard deviation of entropy in clusters for the three constructed graphs.

**Table 5. Clusters genre entropy in graphs.**

<table>
<thead>
<tr>
<th></th>
<th>$G_{NS}$</th>
<th>$G_{En}$</th>
<th>$G_M$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mean cluster entropy ($\mu$)</td>
<td>3.593</td>
<td>2.668</td>
<td>2.952</td>
</tr>
<tr>
<td>Clusters entropy STD ($\sigma$)</td>
<td>0.0901</td>
<td>0.2498</td>
<td>0.2506</td>
</tr>
</tbody>
</table>

Since low entropy indicates less genre diversity in clusters, it can be better from this point of view. In Table 5, the lowest clusters genre entropy is for graph $G_{En}$. This shows that the clusters in graph $G_{En}$ have more similar genres than the clusters in the other two graphs. However, our method has a significant difference with the baseline in this criterion. Also we found by manually checking the results that the reason for the high value of the standard deviation in these three criteria for the MoGaL method is the existence of movies that have a high degree. These movies have common topics that can be close to different genres.

The homogeneity of the $G_M$ with respect to movie genres indicates that movie subtitles are one of the most informative features for discovering similarities among movies. Also our evaluations show that subtitles are a rich and descriptive.
feature for topic extraction and similarity analysis of movies. Genres of movies are usually chosen according to the suggestions of writers and viewers. Therefore, this relationship among the topics extracted by LDA from subtitles and genres indicates that our topics are close to the real world. On the other hand, due to the power of LDA in understanding hidden topics, this constructed graph has information beyond genres, from thematic connections between movies, which are very valuable.

5. Conclusions and Future Work
In this paper, we proposed a method called MoGaL for constructing the graph of the movies based on LDA on their subtitles and evaluating movies' similarity graphs based on their subtitles. In this regard, we first collected the data and subtitles of movies. Then we extracted important topics from the subtitle text using topic extraction methods LDA. In the following, using the cosine similarity measure, we measured the similarity among topic vectors of movies, and based on this, we constructed a movies graph. Next, considering that the genre of a movie represents its dominant topic, we introduced three criteria for evaluating the graph made using homophily and entropy based on the genre of movies and evaluated the graph.

The results obtained show that there is a correlation between the topics extracted with LDA and the genres of the movies. Comparing the results of the constructed graph by MoGaL with two baseline graphs and the graph constructed with the entities of the movies shows that the values of entropy and homophily are very different from the baseline and are very close to the value of the graph constructed with the entities. This shows that the subtitles of movies indicate the genre and content of a movie. Therefore, it can be used as an important feature to analyze movies.

For future work, considering that MoGaL considers the relevance of hidden topics in subtitles, it can be used in the field of genre prediction. This graph can also be used to recommend movies because people usually like movies with similar themes to their favorite movies.

References


ساخت گراف فیلم‌ها با استفاده از LDA بر روی زیر نویس MoGal

محمدرضا حسینی، حسن رحمانی، دادگر مؤمنی و مظهر نصری

دانشکده مهندسی کامپیوتر، دانشگاه علوم و صنعتی، تهران، ایران.
ارسال: 2022/11/01، پذیرش: 2023/3/18

چکیده:
نمایش گراف‌های داده‌ها با روابط بین اجزای داده را بهتر نشان دهد و در نتیجه تحلیل بهتر و غنی تری ارائه دهد. تاکنون فیلم‌ها بارها با استفاده از ویژگی‌های مختلف وی، مانند Zernovs چندانی نشده است، در حالی که Zernovs گراف ها را در بر می‌گیرند و اطلاعات پنهان را ارائه می‌دهد. این مقاله چکیده: