Format intended for *Expert systems with Applications Journal*

Leveraging Semantic Plot Networks for Enhanced Movie Recommendation Systems

Valentina Brivio (3131680), Enrico Cipolla Cipolla (3139480),

Pierluigi Mancinelli (3120534), Arianna Zottoli (3136628)

Abstract

The field of recommendation systems has witnessed remarkable progress over the years, with various methodologies being widely explored in past literature. This study aims at advancing traditional collaborative filtering methods by integrating a network-based structure, extending its application to Movies. We propose a network where films are nodes, connected to one another based on the semantic similarity between the plots' synopsis. The recommendation system is then built on the communities of movies, together with the feature of Genre. On average, users rate the recommended movies higher than the not-recommended. This showcases the validity of our algorithm in providing diverse, accurate, and contextually relevant movie recommendations.

Keywords: Recommender System ; Network Analysis ; Community Detection ; Semantic Similarity

1. Introduction

In the evolving landscape of digital entertainment, effective and strategic movie recommendations become crucial for boosting user satisfaction and engagement. However, traditional recommendation systems, primarily relying on collaborative filtering and content-based methods, often fail to manage the complexity of modern multimedia content. Our study introduces an alternative approach by incorporating semantic networks into movie recommendation systems. We also aim at validating the results obtained by the recommendation system empirically.

In particular, our paper employs a network-based approach where movies are represented as nodes connected by the similarity of their plots. This method builds on and improves existing network-based recommendation systems, which are known for effectively managing complex details of the data. Previous studies, such as those by Son & Kim (2017) and Davoodi et al. (2013), have laid the groundwork by applying network science to various domains, which our research builds upon by focusing specifically on movies.

Our contribution to the literature is twofold. Firstly, we measure the movies' semantic relationships using Natural Language Processing techniques. Secondly, by representing these relationships in a network structure, we detect communities of movies, which make up the basis for our recommendation algorithms. Our approach is therefore quite different from traditional genre-based or user preference-driven models, which do not consider semantic elements.

By leveraging the principles of network theory and the advancements in Natural Language Processing (NLP), we believe that the recommendation system developed in this paper could provide highly accurate, diverse, and contextually relevant recommendations.

2. Literature review

The development of recommendation systems has seen significant advancements over the years, with numerous approaches explored in academic literature.

Recommendation systems are mainly divided into collaborative filtering and content-based recommendation methods. Collaborative filtering uses the known preferences of a group of users to make recommendations or predictions of the unknown preferences for other users. Content-based recommender systems, instead, analyze item descriptions to identify elements that are of particular interest to the user. The goal is to recommend items similar to those that a user has shown interest in the past (Su & Khoshgoftaar, 2009).

Our study builds on traditional content-based filtering by incorporating a network-based approach. In recent years, network-based recommendation systems have gained increasing attention. By incorporating network science principles, the system can better handle the complexity and interconnectedness of item attributes, improving the recommendation performance (Son & Kim, 2017).

The framework proposed in our project is closely related to the methodology outlined in the study by Davoodi et al., 2013. They built a social network based on the semantic similarities between expert profiles in a textual format, containing relevant attributes collected from different online sources. They identify communities within the network using K-means clustering and recommends expert communities or individual representatives to the user based on the semantic similarity of profiles to their information needs. We extend this framework focusing on movies instead of experts, by leveraging the semantic similarity between movie plot synopses to build a network and identify communities of related movies.

This network-based recommendation system aims to provide personalized recommendations to users based on the movies they have watched. This is a type of content-based filtering which uses as attributes the items' semantic similarity (De Gemmis et al., 2015). The items, in our case the movies' plot synopses, are represented as fixed-dimension vectors. The literature on textual item representations is vast, with works focusing on discrete approaches such as TF-IDF (Permana & Wibowo, 2023), continuous approaches such as Doc2Vec (Liu & Wu, 2019), or methods based on the Transformer architecture (Nilla & Setiawan, 2024). We decided to employ the Sentence-Transformer all-mpnet-base-v2 model, an adaptation of the MPNET architecture aimed at creating semantically meaningful sentence embeddings that can be compared using cosine similarity. Reimers & Gurevych (2019) demonstrate the effectiveness of sentence transformers in generating contextual embeddings, which can be used to form edges in a semantic network of items.

A naïve method to generate the edges is to set a threshold, a minimum strength we accept in the network, and discard everything not surpassing the threshold. This strategy can generate problems with real-world networks, characterized by edge weights with highly skewed distributions and correlations, generating too sparse or too dense networks. The disparity filter (DF) addresses these issues by using a node-centric approach, removing less important connections and keeping those that are statistically significant for each node (Coscia, 2021).

For the movies' network construction and community detection, the study by Son & Kim (2017) provides valuable insights. It improves content-based filtering for recommendation systems by using multi-attribute networks. Their approach involves gathering comprehensive item attributes, computing element similarities, and creating a network graph to capture both direct and indirect relationships. In the movie recommender system, a node of the network represents a movie, and a weighted edge represents the degree of relevance between two movies. While they use the number of matching attributes between movies as weights, we use the value of the cosine similarity between the movie plots. They clustered the items using modularity-based methods, which measure the strength of the division of a network into clusters (communities), to form groups that contain users and movies that are closely related (Sibaldo et al., 2014). Finally, they employ centrality measures (degree, closeness, betweenness) to identify key items and bridges through different communities, thereby facilitating diverse and relevant recommendations.

3. Data Collection

Our main source of data was the IMDB Spoiler Dataset, downloaded from Kaggle. The dataset contains 1572 films, with variables like *duration*, *genres* (a list of up to 3 genres for each film), *summary* and *synopsis* of the film. The latter two differ in the amount of information they contain about the movie and its plot. Plot Summary has in fact an average length of 100 words, whereas Plot Synopsis has an average of 1500 circa. Originally, the dataset was meant to be used to train spoiler detection algorithms. In fact, the summary does not contain any information on how the plot evolves, and it is generally shorter, whereas the synopsis is a much more detailed description of the storyline of the movie. We tried to implement both, but we ultimately decided to use the plot *synopsis*, as we expected better results in terms of the recommendation system and higher modularity in community detection. We discuss further the difference among the two in section 7, when testing the recommender algorithm.

Initially, the dataset did not contain the titles, but having the *imdb_Id* of each movie, scraping them from the official Imdb website was straightforward. For the purpose of our analysis, we only kept *title*, *movie-id*, *plot_synopsis* and *genres* of each film.

Additionally, we used the MovieLens 20M dataset for the validation phase. Said dataset contains more than 20 million reviews, with a rating from 1 to 5. We matched MovieLens and the IMDB Spoiler Dataset to select only reviews to movies that were present in our dataset. The resulting dataset consisted in more than one million reviews from 140 thousand different users.

3.1. Preprocessing

The plot synopsis is pre-processed using techniques inherited from NLP. Specifically, we lowered, tokenized and lemmatized the text. We also removed every non-alphanumeric character and removed stop words, a set of commonly used words in a language - such as “a” “the” “is” “are”- that do not carry useful information.

4. Network Construction and Analysis

To offer a more comprehensive recommendation system, we decide to represent the movie as nodes of a network structure, connected with each other according to their pair-wise semantic similarity.

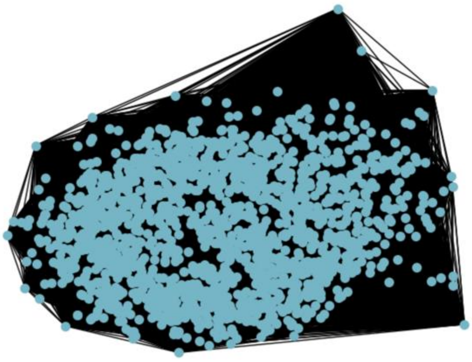

Figure 1: Semantic Network pre-pruning

4.1. Network structure

Indeed, after pre-processing the text, we employed the Sentence-Transformer **all-mpnet-base-v2** model, to produce contextual embeddings of the text. These offer a vectorized representation of the words, maintaining contextual references across different parts of the text.
Furthermore, we used these vectors to produce pairwise cosine similarity scores for all movie synopsis and we leveraged these results to build the graph.
Specifically, cosine similarity shows the direction of similarity between two vectors by measuring the cosine of their angle. The outcome falls between -1 and 1, where -1 denotes complete opposition and 1 shows that the vectors are identical.

We considered the similarity, and we kept only the positive values. This choice allowed us to consider these results as weights for the edges of our network. Hence, as soon as the cosine similarity between two nodes is positive, they are linked through an edge weighted accordingly. However, this method generates an enormous number of connections, most of which having strengths near to zero (Figure 1).

As a result, we explored network backboning techniques to reduce this unnecessary high network density.

4.2. Network backboning

We propose different methods for network backboning:

1. *Naïve method.* As a first attempt, we implemented a heuristic strategy to reduce the number of links in the network. This consists in defining a threshold larger than

zero to yield acceptable network sizes. The defined value was set to be 0.44. However, this approach is not robust enough and it is not grounded on any statistical or network science hypothesis.

2. *Convex network reduction.* This method leverages the properties of convex optimization to reduce the network's complexity while preserving its essential structure. The idea is to formulate the network reduction as an optimization problem where the objective is to minimize a convex function that represents the network's total weight. By guaranteeing that subgraphs are convex, the convex network reduction technique concentrates on locating and maintaining the fundamental structural components of the network. If the shortest path connecting any two nodes in a subgraph stays consistent with the shortest path in the original, larger network, then the subgraph is said to be convex. To do this, the technique considers each non-empty subset of nodes and generates all possible induced subgraphs of the network. Next, we evaluated each subgraph to see whether it was convex. Although this approach is theoretically sound and guarantees the preservation of crucial connectivity properties, it did not provide a true dimensionality reduction, but rather a check of network convexity.

3. *Disparity filtering.* Disparity filtering is a method that reduces the number of links in a network by focusing on the statistical significance of each link. The process involves evaluating the weights of the links in relation to the distribution of weights connected to each node and prune those that do not meet a specified statistical significance level. It maintains the backbone structure of the network and is more computationally feasible than the convex network reduction. However, the results vary depending on the significance level selected, and require the computation of statistical measures and p-values for every edge. Additionally, it makes the potentially erroneous assumption that edge weights follow a normal distribution.

4. *Adjusted disparity filtering.* Adjusted disparity filtering is an enhancement of the basic disparity filtering technique that introduces an adjustment factor to account for the heterogeneity in node degrees and link weights more accurately. Interested in providing a more nuanced assessment of edge significance, this improved method adds a z-score calculation to more precisely account for variations in node degrees and edge weights. Adjusted disparity filtering better captures the significance of each edge by using a two-tailed test to determine the p-value; thus, being more appropriate for networks with high variance in node degrees (scale-free like) and in link weights. This modification enhances the robustness and applicability of the method by generating a reduced network that is more representative.

Ultimately, we chose Adjusted Disparity Filtering and successfully reduced the links to only **8% of the original** number using a p-value of 0.05. The rationale behind this choice is that the method is more applicable to real life networks. Overall, this approach improved computational efficiency whilst preserving the network's fundamental structure, key to the recommender system.

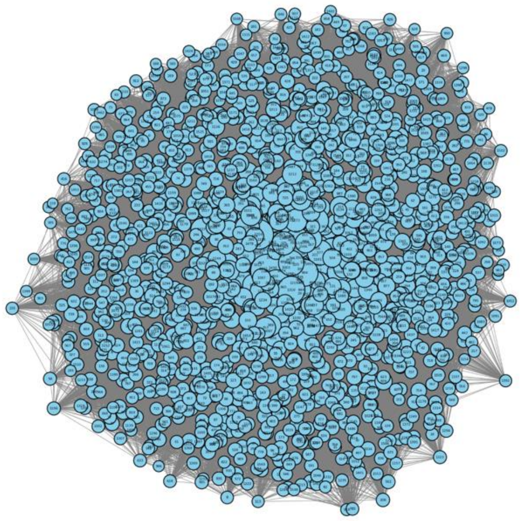

Figure 2: Semantic Network post-pruning

4.3. Descriptive Analysis

In this section we provide some analysis of our network along with centrality measures that describe it, including degree centrality, betweenness centrality, closeness centrality, and eigenvector centrality.

The movie network comprises 1,572 nodes, 106,054 edges, and a density of 8.6%. Regarding the degree, on average, each movie in the network is connected to about 134 other movies. This relatively high average degree indicates that movies in this network are highly interconnected, suggesting that there are many similarities or relationships between the plots of different movies. Moreover, it shows a quite high standard deviation of 108, indicating a wide range of connectivity among the movies. Some movies are much more connected than others, which may suggest the presence of hubs or key movies that are central to the network.
Looking at the degree distribution, instead, the presence of a few highly connected nodes (hubs) and many nodes with low connectivity suggests that the network might exhibit scale-free properties.

Moving to the centrality measures, we obtain:

- *Degree centrality*: measure of the connectedness of each node. The top one is node 276[1] and measures 0.57, signaling that it is connected to 57% of the other nodes in the network.
- *Betweenness centrality*: measure of fraction of shortest paths passing over a specific node. The top one measures 0.0175.
- *Closeness centrality:* measures how close a node is to the others. It is not interpretable per se but the higher the closer to other nodes. The top one is node 276 measuring 0.7.
- *Eigenvector centrality*: measure of the influence of a node in a network. Unlike degree centrality, which only considers the number of direct connections a node has, eigenvector centrality also considers the centrality of the nodes to which it is connected. The top one is node 276 and measures 0.1. This suggests that there are no significant hubs or clusters of highly connected nodes. Indeed, this signals a difficulty in finding highly relevant communities.

[1] Node 276: Title: Il campo,
Genre: ['Drama', 'Thriller']

To conclude, there are high values of degree centrality but low values of betweenness. This means that these top nodes are connected to many other nodes directly, but are not frequently on the shortest paths between other nodes.

It means that these nodes act as local hubs within their immediate neighborhood but do not serve as critical connectors between different neighborhoods or communities within the network. Each node is likely part of a densely connected cluster or community.

5. Community Detection

At the basis of our recommendation system lies the detection of movie communities. We decided to implement different methodologies and then select the one that proved most convenient for our purposes.

5.1. Methodology

Specifically, our research aims at uncovering new communities, not having any original ground truth as baseline. Therefore, we could not leverage the normalized mutual information. Instead, we decided to maximize Modularity [5], as explained by Newman and Barabasi-Albert. Higher modularity should in fact guarantee a better partition of the network into communities. The methods we explored into details are:

- *Hierarchical Clustering - Ward:* this method groups nodes into clusters based on similarity, quantified through a specific distance measure. The distance metric employed to assess the similarity between nodes in this case is the shortest path length. After constructing the distance matrix based on the shortest path length, we reduced it to a condensed matrix to perform hierarchical clustering. We performed hierarchical clustering using the Ward method. The Ward method minimizes the total within-cluster variance, ensuring that nodes belonging to the same cluster are more similar to each other than to those in different clusters. We create a Dendrogram to observe the results after using the Ward method. Then, we cut the Dendrogram at different levels but only keep the best partition, according to modularity maximization. This method outputs 12 communities with a final modularity of 0.17.

- *Louvain Algorithm:* this method is divided in two parts, resembling a classical k-means. Indeed, it first moves the nodes to locally maximize the modularity and it then aggregates the associated nodes into super-nodes. These super nodes form a smaller network, whose edges exist only if there exist at least one node linking two super nodes (communities). As a result, Louvain algorithm's outcome can be visualized using an induced graph, which only represents the super nodes and their links. To conclude, the algorithm iterates over the new, smaller, network to understand whether it can be aggregated further.

We also explored the possibility of applying a divisive algorithm such as Girvan Newman, however, due to the volume of our network, it proved to be excessively time consuming and inefficient. In fact, for large networks its complexity approximates O(N^3), whereas Louvain stabilizes at

$O(N \ln(N))$ being preferred over the previous. Similarly, trials with Infomap – algorithm that uses information criteria rather than similarity – have proven to be unsuccessful. This method aggregated all nodes in only one group, not achieving any relevant community detection.

6. Results

After comparing the methods proposed in the Section 5.1, we applied the Louvain algorithm for community creation since it maximized modularity at 0.28 and it converged faster.
Its application resulted in 6 communities, as shown in Figure 3.

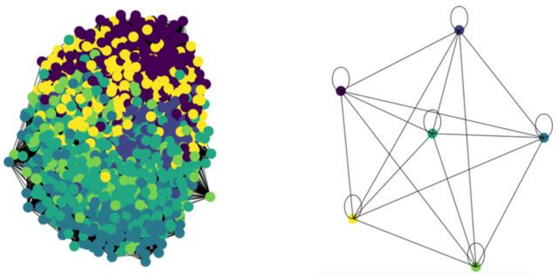

Figure 3: 6 communities in Semantic Network (Left) and induced graph (Right)

Moreover, the edges connecting the different super-nodes have an extremely low weight, proving the strength of the communities and agreeing with the betweenness centralities mentioned in the Descriptive Analysis. From this point on, we always consider communities created with the Louvain Algorithm.

As a last remark, we also acknowledge the limitations of constructing communities according to the principle of maximized modularity. In particular, the Resolution limit. Given the dimensions of the network, it is possible that, to maximize modularity, two or more small distinct communities (having degree smaller than $\sqrt{2L}$) may have been incorrectly merged into one. Further analysis and separation of communities into independent subgraphs might be required to better identify subcommunities. We consider this aspect a possible candidate for future developments, as better community identification may prove crucial to the effectiveness of the recommendation system. Indeed, for future extensions of our research we could also cross validate the resolution hyper-parameter of the Louvain to balance the trade-off between modularity maximization and number of communities.

6.1. Comparison with Movies' genre Communities

We decided to compare the semantic-based network with an alternative network constructed according to Movie Genre. Analogously to the semantic network, the nodes represent the movies, each having as attribute a list of genres. The edges are then created between nodes having at least one genre in common.
The weight associated to each edge measures how many genres the two films have in common (from 1 to 3). The more two movies are similar in genres, the higher the edge's weight. We employed the same method for community creation as for the semantic network, namely Louvain (Figure 4).

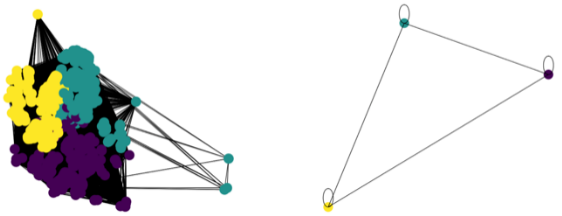

Figure 4: 3 communities in Genre Network and induced graph

The communities created have similar features: despite being 3 in the Genre network, versus 6 in the Semantic network, the modularity is quite similar. The Genre Network has a modularity of 0.279 whereas the modularity of the genre network is of 0.269. We analyzed different metrics to compare the information provided by the two graphs.

- *Normalized Mutual Information (NMI):* A normalized metric that indicates the level of information shared between the two community assignments, giving insights into how similar or diverse the groupings are. Despite being usually employed to compare the performance of different community detection algorithm, we employ it as a measure of overlap between our two results. The metric obtained is of 8.13%, indicating that the two different sets of communities do not share a high percentage of information.
- *Jaccard index:* A metric that compares the similarity and diversity of two sets, in this case, the communities. The semantic communities 4 and 5 appear to be the ones with the least overlap with Genre communities (less than 15%). We find that there seems to be significant overlap for:
  - Genre Community 0 and Semantic community 2 (with an overlap of 29.9%).
  - Genre community 1 and Semantic community 0 (with an overlap of 25.67%).
  - Genre community 2 and Semantic community 3 (with an overlap of 24.40%).
- *Variation of information (VIF):* A metric used to quantify the amount of information lost and gained in changing from one clustering to another. The lower the VIF, the more similar the clusterings are. In our case, the VIF is of 2.61, implying a wide distance of information from the two communities' partition. This metric once again confirms the findings of the NMI.

The comparison highlights the wide difference between the communities generated according to genre and to semantic similarity. This is compliant with our objective of creating communities that generate information different from simple genre differentiation.

7. Recommendation Algorithm

As previously mentioned, the objective of conducting community detection is to provide the basis for the recommendation algorithm. Our recommendation algorithm has been developed with 4 different possible recommendation methods. The first method focuses on selecting the most similar movies, whereas the other 3 methods explore our network in different ways.

7.1. Recommendations 1 (You might like)

In this approach, we select the top 10 films within the community, based on two criteria: the number of genres it shares with the watched film and, in case of ties, the semantic similarity between the films, derived from the weight of the edges in the graph.

7.2. Recommendations 2 (You might try)

This approach ranks these neighbors differently, prioritizing films that have different genres from the watched film while still maintaining high semantic similarity. By focusing on diversity in genres, this method aims to suggest films that are less similar in genre but still closely related in terms of their overall content.

7.3. Recommendations 3 (Expand your horizons)

Instead of focusing on the same community, this algorithm targets films from different communities. It computes the shortest paths from the watched film to all other films in the graph, filtering to include only those outside the watched film's community. The films are ranked based on the length of these shortest paths and their semantic similarity. The idea is to explore connections beyond the immediate community.

7.4. Recommendations 4 (Something Different)

This method aims to diversify recommendations by selecting top films from different communities. After determining the shortest paths from the watched film to other films outside its community, it ranks these films by the same criteria used in Recommendation 3. Then, we group the recommendations by community and select the top N films from each community. This approach ensures that the user receives a varied set of recommendations, representing different communities.

Lastly, we tested and compare the outputs of the algorithm depending on the information used for defining similarity. As mentioned in the dataset description (Section 3), we expected that using plot_synopsis, instead of plot_summary, would lead to better outputs. Plot synopsis is longer and contain detailed information about the movie, regardless of spoilers. This is evidently reflected in the quality of the recommendations. For instance, inputting the movie “Spider-Man”, the algorithm recommended: the other movies of the saga Spider-Man when using plot_summary, and, more broadly, the other movies of the Marvel Saga, when using plot_synopsis (considering for both the Recommendation type 1). We therefore used Plot Synopsis, as it provides a more detailed overview of the storyline, character arcs, and themes. Thus, the recommender has enough information to find deeper connections between movies. This results in a broader and more congruent set of movie recommendations, not just direct sequels, or movies sharing the same key words.

7.5. Prototype

To test our recommendation algorithm, we created a web-app (LINK) using *Streamlit*, an open-source Python framework to create interactive data apps. With the app, you can select which film you want to use as a reference for recommendations, and which one of the recommendation methods you want to use.

8. Validation and Testing

The aim of this section is to prove the efficacy of the recommendation system developed through community detection. In order to do so, we employed the previously mentioned dataset "Movie lens 20M". This dataset contains information about more than 130 thousand viewers, the movies watched, their reviews, and timestamps of each review. Filtering out movies not in our network, the average user has reviewed more than 75 movies.

8.1. Communities and Recommendations Validation

Our validation approach has different steps. First, we wanted to assess the validity of the communities. We tested the hypothesis that users tend to give similar ratings to films within the same community. To validate this assumption, for each user ID we grouped the reviews by community, calculating the standard deviation of ratings. We then compared these intra-community standard deviations to the overall standard deviation of ratings and to the intra-genre-groups standard deviation. The overall standard deviation is 1.03, whereas the average standard deviation in our communities, after filtering for combinations (user-community) with more than 50 films reviewed, was 0.89, and 0.88 for genres groups. This result can be considered valuable considering the large number of observations. Although some variability is still expected, as films with a similar storyline might still be very different, it shows that our communities can be used to group user preferences.
Moreover, we decided to implement a more direct approach, focusing on average ratings. For each user in the Movie Lens 20 M dataset, we considered as "baseline" the first movie with the highest rating granted by the user.

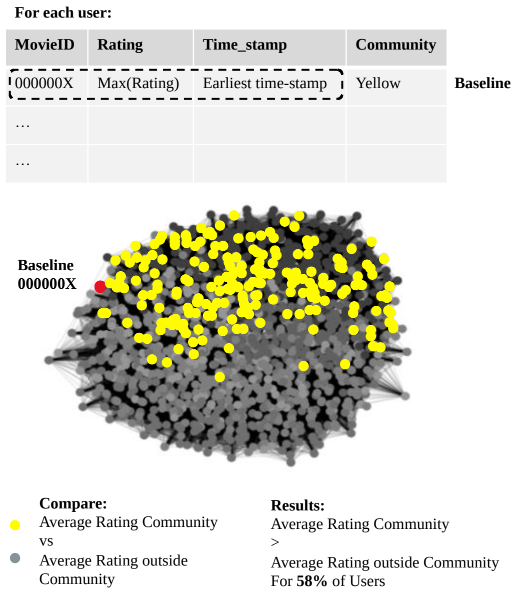

Figure 5: Validation Step 1

The assumption behind the effectiveness of our recommendation system is that the user will be more likely to rate highly the movies in the same community of the baseline. Therefore, we compare the average rating assigned to movies in the baseline community against the average rating assigned to movies outside of said community (Figure 5). The result we obtain is that the average rating inside the community is higher than outside the community for 58% of the users.
Secondly, to further validate our recommendations, we decided to test the first method of our algorithm, namely "You might like".

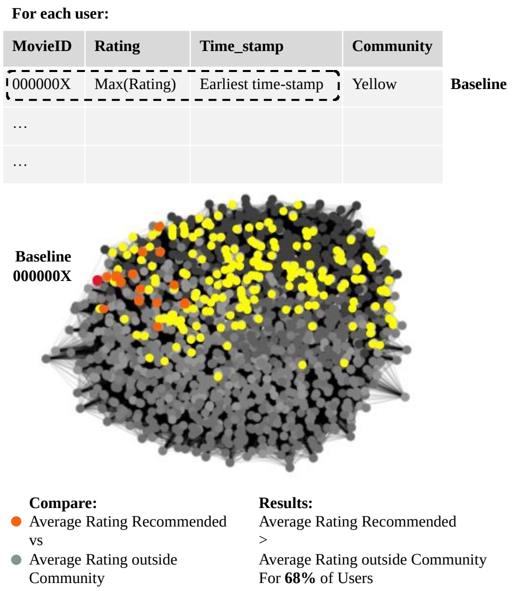

Figure 6: Validation Step 2

After identifying the baseline movie for each user as we did previously, we created through the algorithm the list of 15 movies[2] recommended. Subsequently, we computed the average rating that the user assigned to the movies recommended and compared it to the average rating of other movies watched by the same user (Figure 6). The dataset in this case is much reduced, as it is not certain that the user actually has watched and recorded any of the movies recommended by our algorithm, the observations are in fact reduced to about 70 thousand. However, the results were much improved: the average rating of recommended movies is higher than the one of non-recommended movies for 68% of the users, and on average is higher by 0.25.

9. Conclusions

Our Semantic-based Network system proves to be an efficient method for crafting recommendations. The validation confirms across a wide sample of users that, despite highly variable personal preferences, on average, users will rate the movies recommended higher than those not recommended. This is an important result, as it validates the effectiveness of communities for the construction of our recommender system.

Our recommendation system is designed to enhance user discovery and satisfaction by providing diverse and engaging recommendations. Unlike traditional models that typically focus on sequels, star notoriety, or genre similarity, our system exploits semantic similarities drawn from synopsis analyses. This approach not only broadens the recommendation scope but also significantly boosts the visibility of niche or lesser-known films, often overshadowed by mainstream movies. For instance, by analyzing thematic and narrative elements, our network can connect a popular movie like 'Inception' to an indie film that shares similar conceptual themes, and introducing audiences to films outside their usual viewing habits. This method effectively reduces bias towards high-budget films, offering a more equitable recommendation system. Furthermore, the network structure allows for the identification of optimal paths that strategically lead viewers from highly watched movies to in-house productions, maximizing streaming potential for target

[2] The algorithm "You might like" typically picks 10 recommended movies. To limit the cases of users not having seen the recommended movies, we expanded the recommendation set to 15 movies in this case only. This allows us to have a wider set of valid observations.

titles without compromising on relevance or quality of recommendations. This technique could prove particularly beneficial for streaming platforms looking to promote their original content.

Finally, the methodologies developed could be adapted for other types of media content recommendations such as books, music, or even video games, where narrative and thematic elements play a crucial role in user preferences. In conclusion, this study presents a significant advancement in recommendation systems by integrating network-based structures into traditional collaborative filtering methods, specifically applied to movie recommendations.

9.1. Limitations and Future Developments

Our methods also present a set of limitations. For instance, it does not take user similarities into account and rely on the similarity between the plot synopsis only. Further validation might become necessary, for instance by testing the algorithms through surveys.

In addition, for future development we are considering the integration of additional features to refine the similarity metrics used in the algorithm. Specifically, we plan to enhance the personalization of user recommendations by allowing the system to dynamically adapt to changes in user preferences over time. In addition, we could expand the similarity score between two films by including a linear combination of different parameters. The first option would be considering the number of actors shared between two films, which could provide insights into casting similarities that might appeal to certain audience segments. Additionally, we could also leverage external user-generated content, such as film recommendations shared on blogs and forums (e.g. Movie Recommendations Subreddit). By analyzing instances where two films are frequently recommended together, or sequentially, we can infer a deeper level of thematic or audience congruence.

References

[1] Coscia, M. (2021). The atlas for the aspiring network scientist. arXiv preprint arXiv:2101.00863.

[2] Davoodi, E., Kianmehr, K., & Afsharchi, M. (2013). A semantic social network-based expert recommender system. Applied intelligence, 39, 1-13.

[3] De Gemmis, M., Lops, P., Musto, C., Narducci, F., & Semeraro, G. (2015). Semantics-aware content-based recommender systems. Recommender systems handbook, 119-159.

[4] Liu, G., & Wu, X. (2019, March). Using collaborative filtering algorithms combined with Doc2Vec for movie recommendation. In 2019 IEEE 3rd Information Technology, Networking, Electronic and Automation Control Conference (ITNEC) (pp. 1461-1464). IEEE.

[5] Newman, M. E. J. (2006). "Modularity and community structure in networks". Proceedings of the National Academy of Sciences of the United States of America. 103 (23): 8577–8696.

[6] Nilla, A., & Setiawan, E. B. (2024). Film Recommendation System Using Content-Based Filtering and the Convolutional Neural Network (CNN) Classification Methods. Jurnal Ilmiah Teknik Elektro Komputer dan Informatika (JITEKI), 10(1), 17-29.

[7] Pazzani, M. J., & Billsus, D. (2007). Content-based recommendation systems. In The adaptive web: methods and strategies of web personalization (pp. 325-341). Berlin, Heidelberg: Springer Berlin Heidelberg.

[8] Permana, A. H. J. P. J., & Wibowo, A. T. (2023). Movie Recommendation System Based on Synopsis Using Content-Based Filtering with TF-IDF and Cosine Similarity. International Journal on Information and Communication Technology (IJoICT), 9(2), 1-14.

[9] Reimers, N., & Gurevych, I. (2019). Sentence-bert: Sentence embeddings using siamese bert-networks. arXiv preprint arXiv:1908.10084.

[10] Sibaldo, M. A., de Carvalho, T. B., Ren, T. I., & Cavalcanti, G. (2014). Recommender system based on modularity. In Proceedings of the 2014 Recommender Systems Challenge(pp. 58-61).

[11] Son, J., & Kim, S. B. (2017). Content-based filtering for recommendation systems using multiattribute networks. Expert Systenon ms with Applications, 89, 404-412.

[12] Su, X., & Khoshgoftaar, T. M. (2009). A survey of collaborative filtering techniques. Advances in artificial intelligence, 2009.
[13] Yu, F., Zeng, A., Gillard, S., & Medo, M. (2016). Network-based recommendation algorithms: A review. Physica A: Statistical Mechanics and its Applications, 452, 192-208.