r/marvelstudios: Community and Content Analysis after the release of Avengers: Endgame

Roberto Ferrari 852220, Davide Prati 845926, Marco Sallustio 906149

Abstract
This report analyzes interaction within the r/marvelstudios subreddit, focusing on the discussion of Avengers: Endgame, one of the most popular films in the Marvel Cinematic Universe (MCU). With the objective of identifying relationships, influential users, sentiments, and communities, the research employs Social Network Analysis, Sentiment Analysis and Topic Modeling. The report's structure comprehends data collection and exploration, social network analysis, sentiment analysis, and topic modeling. The approach is centered on understanding dynamics within an online community engaged in passionate discussions on a popular topic like Avengers: Endgame.

Keywords
Graph Theory - Community Detection - Sentiment Analysis - Topic Modeling - Reddit - Social Media Analytics

Contents

Introduction

Reddit is a social media platform based on content sharing, discussion, and community voting.
It is organized into 'subreddits', which are thematic communities created by users. Each subreddit focuses on a specific topic, such as news, hobbies, or entertainment.
Users, commonly referred to as 'redditors', can subscribe to one or more subreddits based on their interests, allowing them to view and participate in discussions within those subreddits. They can engage by creating posts or commenting on existing posts.

In particular, we decided to analyze the subreddit r/marvelstudios, that, at the time of writing this report and conducting the project, with 3.5 million members, it ranks 168th among the world's most popular communities [1]. This subreddit is dedicated to discussing Marvel Studios' films and series and anything else related to the Marvel Cinematic Universe (MCU). We chose this subreddit because this is a rather popular topic among young people and therefore very popular in social networks like Reddit; furthermore, the selection of this subreddit was driven by the fact that Marvel movies often elicit diverse and controversial opinions among users, leading to intriguing discussions that would be particularly interesting to analyze.
For this reason we decided to analyze the interactions generated for one of the most popular Marvel films: Avengers Endgame.
This film in fact marks the end of Marvel's epic Infinity Saga and achieved great success with critics and audiences, setting numerous box office records, also becoming the highest grossing film in the history of cinema.

Objective

Based on the considerations made previously we have identified some of the main objectives of our analysis:

- Identify the relationships generated within the subreddit and more specifically within our discussion of interest;
- Identify the most influential users through accurate Social Network Analysis;
- Understand the feedback that the users of the subreddit have had through the sentiment generated by each one;
- Visualize the main topics discussed in the community

and highlight groups of words that tend to relate to a specific topic rather than another;

- Identify any communities present and their characteristics.

Structure of the report

The report is organized as follows:

1. **Data Collection**: in this stage, we accessed the Reddit API through PRAW to obtain the dataset we will be working on. This dataset will undergo a preprocessing phase, during which we will apply various steps to make it suitable for the purposes and models we intend to apply it.
2. **Data Exploration**: in this section, we present some graphs and visualizations that provide an overview of the construction and characteristics of the preprocessed dataframe.
3. **Social Content Analysis**, divided in
   - **Sentiment Analysis**: section in which we perform sentiment analysis on the subreddit, classifying comments as positive, negative, or neutral using specific scores to assign each comment to the correct category. The section also includes analysis and visualizations of the dataset, this time divided into the three types of sentiment, compared against each other.
   - **Topic Modeling**: after a preparation phase, we used the Latent Dirichlet Allocation (LDA) algorithm to compute topic modeling. To improve interpretation, we represented the results visually through various figures.
4. **Social Network Analysis**: We conducted a social network analysis, creating a graphical representation of the network, calculating and studying its associated metrics. Additionally, we computed the modularities of the communities, based on which we performed community detection on the six most significant ones.

1. Data Collection

We chose to use PRAW to extract our data from Reddit. PRAW (Python Reddit API Wrapper) [2] is a Python library that simplifies access to the Reddit API.
With PRAW, you have the capability to develop Python scripts and applications for interacting with Reddit data. This includes tasks such as reading posts, posting comments, retrieving user information, and a variety of other functionalities.

Furthermore, we decided to create our dataset by extracting the comments for each post in order to obtain as much data as possible and therefore to more easily extract discussions regarding our topic of interest. To obtain useful comments for our analysis, we performed a search for posts in the "marvelstudios" subreddit that contain the keywords 'avengers' or 'endgame'. We thus obtained a dataset containing 112 567 columns.
To avoid having results that were too sparse and not very relevant to our analysis, we filtered our dataset so that it only contained comments created between April 26, 2019 and April 26, 2020; therefore starting from the release date of the film in the USA for one year, obtaining a final dataset of 67 764 rows.

We decided to extract for each comment the following features:

- author_submission: author of the post
- title_post: title of the post
- score_post: score number of votes received by the post, indicates the "popularity" of the post
- url_post: URL of the post
- created_utc_post: date and hour of the creation of the post
- author_comment: author of the comment
- body: text of the comment
- created_utc_comment: date and hour of the creation of the comment
- score_comment: score number of votes received by the comment, indicates the "popularity" of the comment

1.1 Data Preprocessing

Before proceeding further, we applied a significant preprocessing phase to the data obtained and stored in our dataset, in order to make the extracted text as usable as possible for the algorithms and models we intend to apply.
For this purpose, a series of operations were carried out, mainly conducted through the use of regex rules.
The operations are as follows:

- Lemmatization;
- Removal of links;
- Removal of e-mail addresses;
- Removal of numbers;
- Removal of extra-white spaces;
- Removal of punctuation;
- Removal of emoji;
- Transforming text to lowercase;

We would like to point out that all the operations listed have been applied to all comments with text in English, excluding all other comments written in other languages.
To do this we created a specific function designed to detect the language of a text using the pycld2 library in Python [3]. Essentially this function uses the pycld2 library to detect the language of a text and returns a language identifier.

Furthermore, before doing the Stemming operation we also decided to manage the contractions, which are very frequent in the English language. For example, the sting of characters “I’m” was always transformed into “I am”, or “I’ve” was transformed into “I have”. This was done because during the stemming operation these contractions could have caused some problems.

2. Data Exploration

After the completion of the preprocessing phase, our dataset is ready to undergo an exploratory analysis in which we visualize some significant elements and features: in particular, this is going to be the first of two exploratory phases, where we will focus on the dataset before the sentiment analysis is performed, considering primarily the relevance of the topic (i.e. the film itself) through time.
It is important to remind that the information retrieved about the topic refers to the period of time subsequent to the release of the “Avengers: Endgame” film, in order to obtain the actual sentiment of the users, without factors such as the hype surrounding the culminating chapter of the Avengers saga, which would have constituted obvious bias, and, of course, a skewed representation of the sentiment.
Firstly, we report the popularity of the topic by observing the number of comments under submissions that talked about it through time: as we can see, the highest peak in the subreddit is registered with 15 202 comments, corresponding to the first days after the official release of the film (2019/04/24), then drops significantly a week later.

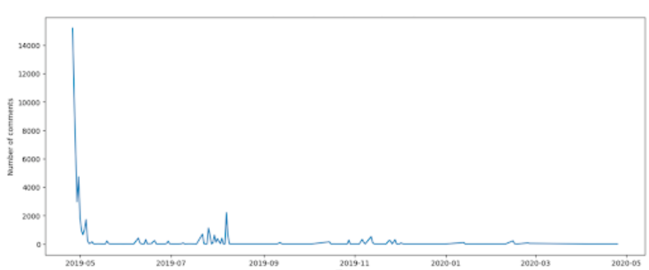

Figure 1. *Avengers: Endgame popularity through time*

The topic seems to return slightly discussed, not as much as at the start though, months later, in August, with a peak of 2213 comments: this amount coincides, almost completely with the exception of one comment, with the responses under a submission posted by the Russo brothers, who were the directors of the film.

Then, we performed a Word Cloud analysis, in order to inspect the most frequently used words: among the most picked ones we found words related to the main characters of the series, such as ’Thano’ (7497 times), ’Tony’ (5583 times), ’Thor’ (5445 times) and ’Hulk’ (3018 times), other ones related to particular elements of this final chapter, for example ’stone’ (4077 times) and ’timeline’ (3468 times), and, of course, some obvious ones such as ’movie’ (10 217 times) and ’Endgame’ (4137 times)

Figure 2. *WordCloud on the preprocessed text*

3. Social Network Analysis

In this section we present a social network analysis of the subreddit, i.e. the process of investigating social structures through the use of network and graph theory, that can capture the dynamic interactions between commenters and post authors.
In this graphical representation, each instance of a comment resulted in the formation of an edge connecting the commenter to the original author of the content being commented upon. The strength of this connection was intricately determined by the score assigned to the comment, thereby creating a weighted, and directed multigraph.
It should be noted that we have excluded from the creation of the graph all edges with a weight lower than 0 since in a weighted graph the use of negative weights could introduce complications in the representation of the graph and in the definition of some graph analysis measures.
This network was comprised of a 25 388 distinct nodes, representing authors, intricately interconnected by 32 747 edges.

3.1 Graph

To create the graph of the network, to highlight interactions and relationship we exploited the use of Gephi [4], a specific open source software for the analysis and visualization of social networks.
To visualize the graph in the best possible way, the OpenOrd layout was used. This layout has a parameter called Edge Cut which controls the maximum length of an arc during the

optimization process. The default value is 0.8: this means that, during the optimization process, every arc with a length greater than 80% of the length of the longest arc present is cut. This allows clusters to separate, as long edges with high weight may have too much influence on distant clusters.

The figure below shows the graph in which we decided to adjust the size of the nodes and labels (name of the account associated with the node) based on the weighted degree of each nod, which is the sum of the weights of the incoming and outgoing edges. In practical terms, weighted degree is useful for understanding the overall amount of 'influence' or 'strength' of a node in the network, taking into account the strength of its connections.

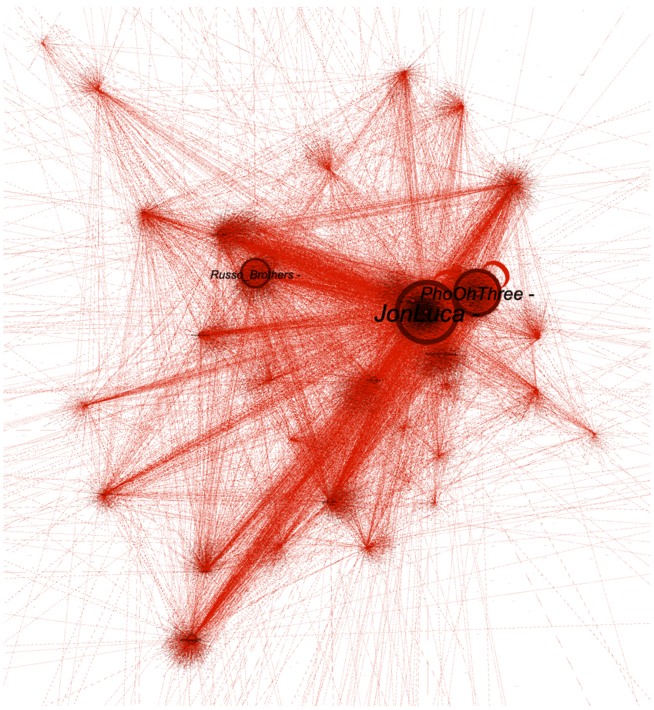

Figure 3. *Graph regulated by the weighted degree*

3.2 Metrics

At this point, we proceeded by calculating the values of some standard metrics regarding the graphs, applying them specifically to our graph. This provides us with important information about the nature of our network. In the table below, we present the values obtained for these metrics.

Metric	Value
Density	0
Avg. Degree	1,29
Avg. Weighted Degree	88,25
Avg. Path Length	1,764

Table 1. *Graph metrics*

Analyzing these results, regarding first of all the density of the graph, it is is equal to 0. This value indicates that there are very few connections in the network compared to the total number of possible connections; so the network can be clearly categorized as sparse.

Continuing the analysis, the average degree represents the average of the number of incoming and outgoing connections for each node in the network. In this case, it appears that on average each node has about 1.29 connections. This relatively low value could indicate a not very dense network, and this result is consistent with the value we found for the density.

The weighted average degree takes into account the weights of the edges in the network. This value suggests that, on average, each node has a much higher weighted sum of connections (88.25) than the unweighted degree (1.29). This could indicate that some connections are much stronger than others in the network.

Finally, the average path length is defined as the average of the shortest distances between all pairs of nodes in the network. In this case, the average path length is 1,764 edges, which indicates that the paths between nodes in the network are, on average, relatively short.

In order to understand who the main influencers of the network are, we computed two centrality metrics: Closeness Centrality and Betweenness Centrality.
These measures allow us to find the nodes that take on an important structural role within the network: the first provides information on which actors are the most central, therefore with the shortest distance from the rest of the nodes; the second instead shows which users, if any, are useful for connecting different parts of the graph and potentially putting different communities in contact.

We have reported in Table 2 the 10 nodes with the highest value of Betweenness Centrality:

User	Betweenness Centrality
The_Asian_Hamster	14530.1
KostisPat257	13641.5
Flamma_Man	9227.5
Sisiwakanamaru	4520.5
PhoOhThree	3600.6
kahlkorver	1983
MangoJam18	1294
LordHyperBreath	670.9
dannys717	570
ENusatron	392.08

Table 2. *Betweenness centrality*

As regards the closeness centrality values, we obtained many nodes with a value of 1; more precisely we have that 64% of the nodes have a value equal to 1 which represents the maximum value that a node takes on in our network.
It would not make sense to report a table with the 10 nodes with the highest value because they would all have a value equal to 1 and for this reason we decided to observe and report in the Table 3 the Closeness Centrality values for the same nodes present in Table 2 in order to make a comparison between the two values:

User	Closeness Centrality
The_Asian_Hamster	0.58
KostisPat257	0.85
Flamma_Man	1
Sisiwakanamaru	0.42
PhoOhThree	1
kahlkorver	0.5
MangoJam18	0.61
LordHyperBreath	1
dannys717	1
ENusatron	0.66

Table 3. *Closeness centrality*

Therefore, comparing the values within the two tables we can make several considerations:

- Users like The_Asian_Hamster, KostisPat257, and Flamma_Man are central in terms of both betweenness and closeness. These users play key roles in network connectivity and are quickly accessible to others.

- Some users show significant differences between betweenness and closeness values. For example, Sisiwakanamaru has a relatively low betweenness value but a lower closeness value, suggesting that it may be less involved in key pathways but is well connected with neighbors.

- Users like ENusatron and MangoJam18 could be considered more peripheral in the network, as they have relatively low values of both betweenness and closeness.

- Sisiwakanamaru and MangoJam18, with relatively low closeness, may be more isolated than other users.

- Users with high betweenness, such as The_Asian_Hamster and KostisPat257, could play mediation and bridging roles, while those with high closeness, such as Flamma_Man and PhoOhThree, are quickly accessible and could have a more direct impact on information dissemination.

3.3 Community Detection

With the aim of analyzing the communities of interest, the modularity value of our network was calculated. Modularity is an important measure of network or graph theory. It was designed to measure the strength of the division of a network into modules (also called groups, clusters or communities). Networks with high modularity have dense connections between nodes within clusters but sparse connections between nodes in different clusters. Gephi to calculate the modularity value uses the Louvain algorithm [5], one of the fastest algorithms available, considered cutting-edge for its high performance.
We found 485 communities with a modularity value of 0.744. This high value of modularity tells us that within the network there are dense connections between nodes belonging to the same cluster, but very sparse connections between nodes that are part of different clusters.
We show below the Community Detection graph in which only the 6 most significant community classes have been selected and displayed, i.e. those that contain the greatest number of nodes (the most populous). The 6 communities are thus shown below with the community number and associated color:

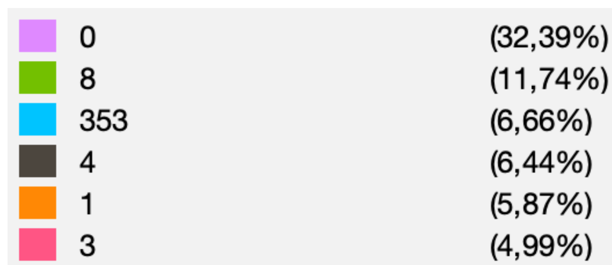

Figure 4. *The 6 most significant communities*

Also in this case the size of the nodes was adjusted based on the weighted degree and furthermore each node is colored differently based on the color associated with the community to which it belongs:

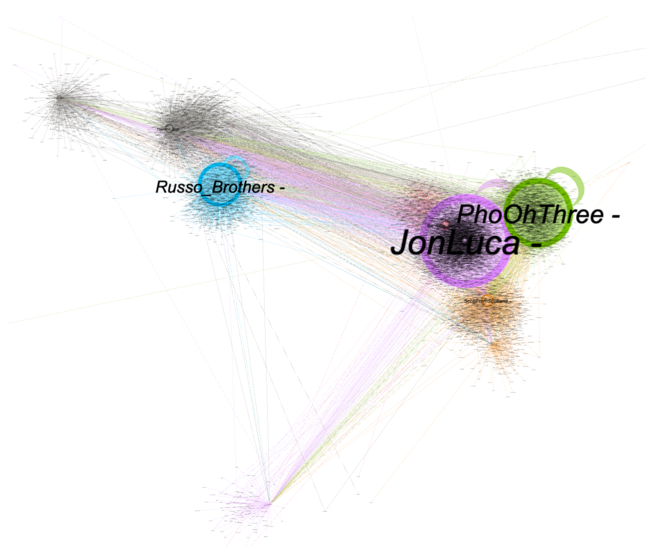

Figure 5. *Community detection*

Thus, we observe that the communities with a higher population correspond to nodes with larger dimensions, as indicated by the weighted degree value. This suggests that there may be a relationship between node size (based on weighted degree) and its membership in a community. One reason is certainly that more central nodes with a higher degree might have a greater probability of belonging to larger communities, node centrality can be reflected in the weighted degree, and central nodes are often important to the overall structure of the graph. Another rationale is that communities could grow around central nodes with a higher weighted degree, as these nodes attract more connections over time and indeed some nodes with a higher weighted degree could play a key role in maintaining the integrity and cohesion of communities, leading to the formation of larger communities around them.

Finally, to better analyze the communities found, we report the 6 communities and the associated sentiment values in the figure below to understand whether sentiment influenced the division of the communities:

Community	Populousness	Positive	Negative	Neutral
0	8131	49,61%	33,20%	17,10%
8	2948	46,80%	32,50%	20,65%
353	1673	61,92%	19,78%	18,29%
4	1817	48,67%	32,40%	19,35%
1	1474	47,15%	33,24%	19,60%
3	1253	47,72%	33,28%	18,99%

Figure 6. *Communites' sentiment*

As you can see, we have very similar sentiment values between these 6 communities; this certainly indicates that people within each community interact similarly, with a balance between positive, negative and neutral comments. This could be a sign of cohesion and coherence within communities.
The only different values are for community 353 which has a higher positive sentiment value (almost 62%) and more unbalanced than the other communities. In fact, observing the graph in Figure 3 we know that this community includes the user 'Russo_Brothers' whose node is in fact positioned in a more distant position than the others.
This may be because this community may cover topics or content that is significantly different from other communities. This diversity in content could influence sentiment more markedly, leading to a more distant arrangement in the layout we chose to use.

4. Social Content Analysis

4.1 Sentiment Analyisis

Sentiment Analysis was performed to detect the polarity (positive, negative, neutral) of comments' texts.
The model used is VADER (Valence Aware Dictionary for Sentiment Reasoning) [6], that is sensitive to both polarity (positive/negative) and intensity (strength) of emotion. More precisely, it relies on a dictionary that maps lexical features to emotion intensities, known as sentiment scores. The sentiment score of a text can be obtained by summing up the intensity of each word in the text, which ranges between -1 and 1. A score close to -1 indicates a very negative sentiment, a score close to 1 indicates a very positive sentiment, and a score close to 0 indicates a neutral sentiment.
Obviously this sentiment analysis phase was carried out after the pre-processing phase explained in the section 1.1.

To have even more detailed values, still using the VADER library, we have created three different functions for each polarity (positive, negative, neutral) which return the sentiment score rather than a compound score.
So, for example, for the first row of the dataset we obtained this scores which we interpret in the following way:

- **CompoundValue**: -0.9745
  The compound score is very negative, indicating that the first sentence has a strong negative sentiment. Since it is close to -1, this is a very bad rating.

- **NegativeValue**: 0.328
  The negative sentiment score is 32.8%, indicating that a significant portion of the sentence has a negative tone.

- **PositiveValue**: 0.065
  The positive sentiment score is 6.5%, indicating that only a small part of the sentence has a positive tone.

- **NeutralValue**: 0.607
  The neutral sentiment score is 60.7%, indicating that the majority of the sentence has a neutral tone.

After doing this, we decided to create a new column in our dataset that directly contained the sentiment value associated with the comment (Positive, Negative or Neutral). To do this we used the CompoundValue obtained previously and we established a specific threshold in order to convert the numerical value into the associated sentiment. We then analyzed the distribution of the sentiment generated for our dataset and obtained that:

- The percentage of negative comments is 31.62%

- The percentage of positive comments is 49.61%

- The percentage of neutral comments is 18.77%

This result is also displayed graphically in the following histogram:

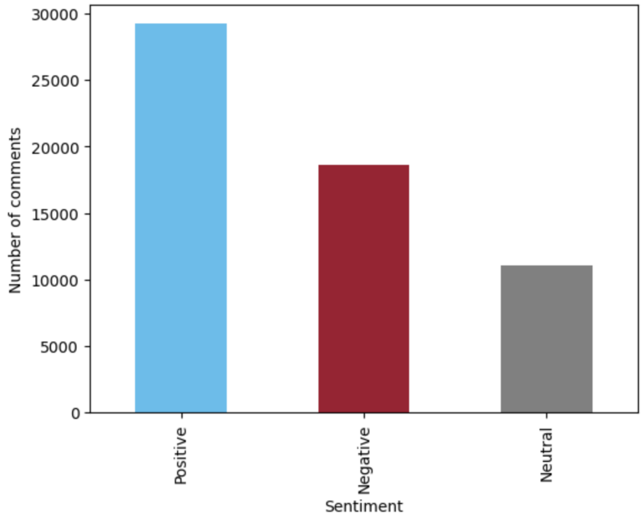

Figure 7. *Sentiment distribution*

So, it seems that the much anticipated final chapter of the Avengers saga was appreciated by the majority of the r/marvelstudios community.
One interesting aspect is related to the length of the comments, i.e. the number of words, observed for each sentiment label. To highlight this, we created three violin plots, that represent the three sentiment labels, and excluded outliers with the interquartile method, which provided a much clearer visualization of the plots.

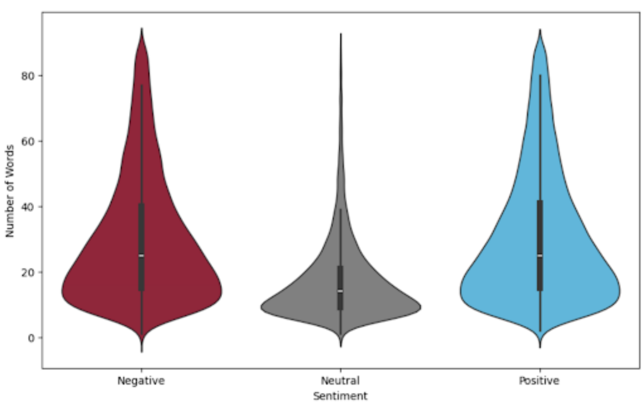

Figure 8. *Violin plots on sentiment distribution*

As we can see the length is not identical for all sentiment labels and, surprisingly, the violin plot referred to the neutral sentiment is more compressed than the other two, meaning that the verbosity of this type of comments is more contained than the positive and negative ones: this is unexpected, since it is safe to assume that a neutral comment would require a more elaborate explanation of why our experience was not primarily good or bad, rather then straight out a positive or negative comment, which can be more direct and concise when expressing the appreciation, or the disapproval, of something we experienced.

Now, leveraging this additional column, we were able to analyze the weekly time trend of the number of comments categorized according to sentiment (positive, negative, and neutral). The result is displayed in the following figure:

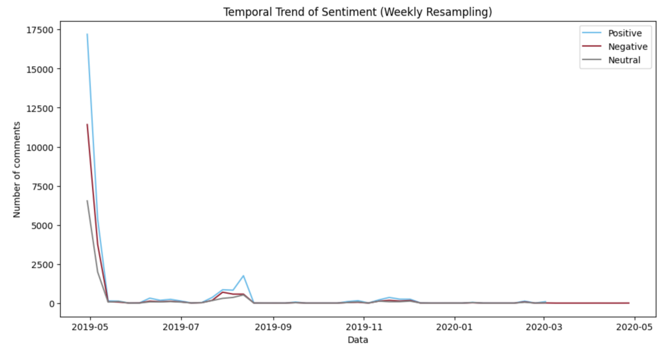

Figure 9. *Temporal trend of sentiment*

From the figure it is possible to see how in the first week of the movies's release there are obviously many comments with a majority of positive sentiment, as has already been noted before. As time passes, on the other hand, the number of comments obviously decreases until reaching the weeks between August and September in which there is a sudden increase in the number of comments in which positive sentiment is always predominant. This sudden increase could be due to the fact that the movie was made available again in cinemas during this period with the addition of some extra scenes.

To also obtain further information on sentiment, we created two different Wordclouds for positive sentiment and negative sentiment, to understand which words are predominant in the two different sentiments.

Figure 10. *Positive WordCloud*

From Figure 10 among the different words we note the names of some of the main protagonists and the most loved superheroes, such as 'cap', 'tony' or 'thor' and also other words that certainly express a more than positive sentiment such as 'love'.

Figure 11. *Negative WordCloud*

From Figure 11 instead we note as the first word 'thanos', who is the main antagonist of the film; or it is possible to notice the presence of other words that certainly express a negative sentiment such as 'die', 'destroy' or 'kill'.

4.2 Topic Modeling

Topic modeling is an unsupervised Machine Learning technique that automatically identifies and extracts significant topics from a set of documents. By leveraging this technique, we can determine the optimal way to categorize various comments based on their main topic. This allows us to cluster the comments into k clusters, where k is chosen to optimize certain parameters depending on the respective topic.
The fundamental concept involves clustering words into distinct groups, where:

- Each word in the cluster is likely to occur "more" (have a probability of occurrence) for the given topic;
- Different topics have their respective clusters of words along with corresponding probabilities;
- Different topics may share some words and a document can have more than one topic associated with it.

Thus, topic modeling can be considered as a soft clusterization.

In this study, to perform topic modeling we exploited the Latent Dirichlet Allocation (LDA) algorithm, implemented using the gensim library [7].

The determination of the number of topics was assessed through the metrics `U_mass` and `C_v` coherence. The `U_mass` measures the degree of coherence of the topics by assessing the semantic similarity between words within the same topic: a lower value (close to zero) indicates greater coherence, while `C_v` measures the coherence of arguments by assessing the pairwise similarity between the main terms of an argument. Thus, a higher value indicates more interpretable topics.
In addition to quantitative metrics, our assessment of model performance incorporated qualitative measures, such as human judgment. Given that these metrics did not have a strong correlation with human interpretability [8], we improved our analysis by visualising the topic models using Worldclouds and different visual figures. This made it much easier to examine in detail the top terms most closely associated with each topic, providing a deeper understanding of the content and context of the topics.

4.2.1 Text Representation

We now focus on the text representation phase of the dataframe to fit it with the required parameters for the LDA model. However, before doing so, we have removed stop words that would otherwise dominate our analysis without providing any additional meaning or discrimination between topics. To achieve this, we used the list of English stop words, excluding the word 'not,' which is useful to retain for the bigrams and trigrams we will create later. Additionally, we added the words 'rimosso' (removed) and 'cancellato' (deleted), which may appear in the dataset but clearly do not refer to the original text created by the user but rather to internal Reddit operations.

Then, to finally represent the text, we used the Bag of Words approach, starting with the construction of bigrams and trigrams to capture more intricate relationships and structures among words in a text than individual words. Bigrams were defined as pairs of words that occurred at least 200 times in the corpus, with a normalized pointwise mutual information (npmi) score of at least 30%. Trigrams, on the other hand, were composed of words that appeared at least 50 times with an npmi score of 0.3%. Once identified, they were added to the corpus, as we wanted to retain, for example, both the words 'battle' and 'hero' and the bigram 'battle_hero'. A total of 1047 bigrams and 59 trigrams were identified. Then, for the implementation of LDA algorithm we created a dictionary and and filtered it by excluding words that appear just in less than 3 documents or in more than 15% of the total number of documents. As a result, a dictionary of 10 238 words was obtained.
Then we employed a bag-of-words corpus transformation. This is consistent with the observation that LDA works optimally when a raw count of word frequencies is presented.

4.2.2 The LDA model

Latent Dirichlet allocation (LDA) is a generative statistical model that allows us to discover hidden topics from a collection of documents. Each document is assumed to contain a mixture of these hidden topics, and each topic is assumed to generate words according to its own probability distribution. The goal is to identify a set of topics that are most likely to generate the observed document data.
To accomplish this, we need to determine the ideal number of clusters (k) into which we should divide the documents, corresponding to the topics we want to consider. The `C_v` and `U_mass` coherence, as discussed earlier, come to our aid for this task. We executed the model multiple times, varying the number of topics from a minimum of three to a maximum of nine. Ultimately, we chose to develop the model with k=7 topics due to the sufficiently high values of both parameters

observed in that case, which are 0.355 and for C_v and -4.106 for U_mass.
Of course, it is better to underline that this choice was solely based on our judgment and a balance we struck between the goodness of the two parameters. Other choices would have been possible with different approaches.

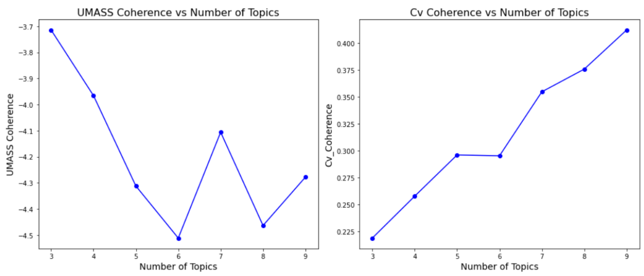

Figure 12. *C_v and U_mass values for the LDA models*

Thus, at this point, we proceeded with the model configured with the number of topics set to 7.

4.2.3 Results

To visualize the results, fist of all ee generated an interactive graph called an Intertopic Distance Map, which illustrates the sizes and relationships between various clusters/topics. Below is an image showcasing the initial visualization, but interactively, by navigating through the notebook, one can explore the top 30 most prevalent words in each topic relative to their overall frequency in the text.

Figure 13. *Intertopic distance map*

But a more significant visualization for our purpose is undoubtedly that of the WordClouds, that we generated based on the word weights for each topic. The weights were computed using the γ parameter, set to 0.6, which is used to balance two factors: the probability of a term within a topic against its overall probability in the corpus and the overall probability of the term in the corpus in the pyLDAvis library [9].

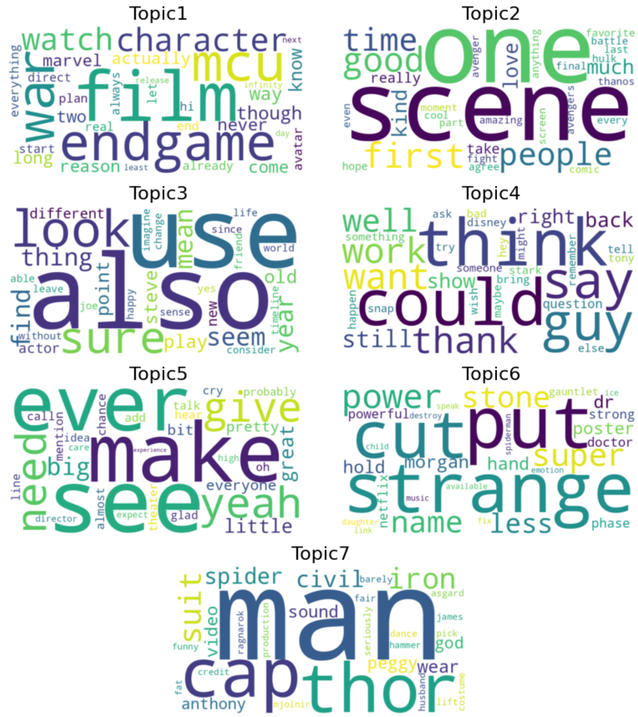

Figure 14. *The WordClouds obtained for the 7 topics*

At this point, having the representations of the most frequent words in each topic, we can try to understand what distinguishes one from another and give them a brief explanatory title, always based on our judgment. We proceeded in the following manner:

1. **Topic 1**: general opinions on the movie, expectations,
2. **Topic 2**: focus on particular memorable scenes, and favorite moments,
3. **Topic 3**: actors and characters' development, differences between this film and the previous ones,
4. **Topic 4**: critical reflections on the plot, actions, and thoughts of the characters,
5. **Topic 5**: astonished and sensationalistic comments on impactful moments and unexpected plot twists,
6. **Topic 6**: specific characters and their superpowers, with reference to a cut scene, likely featuring Dr. Strange as its protagonist,
7. **Topic 7**: focus on the most iconic chararcters: Thor, Captain America, Iron Man.

Conclusions and future developments

At the conclusion of this project, we can say that we managed to achieve the various objectives we had set ourselves.

In particular, thanks to social network analysis techniques we were able to identify the main players in the discussion by first analyzing the betweenness centrality and then the closeness centrality; we thus obtained very high values for the two metrics which showed us how many nodes within our network are crucial nodes for the discussion.
Furthermore, we also met one of our goals regarding the need to find how our network is divided into communities; we obtained that in the largest communities found, all the nodes with a higher weighted degree value corresponded. This phenomenon leads us to support the thesis that central nodes with a high degree of weighting could have a greater probability of belonging to larger communities and that communities could grow around central nodes with a higher degree of weighting, since these nodes attract more connections over time and indeed some nodes with a higher degree of weighting could play a key role in maintaining the integrity and cohesion of communities, leading to the formation of larger communities around them.
We can say that we also managed to satisfy the objective in which we asked ourselves to understand user feedback regarding the analyzed topic of interest; to do this we used sentiment analysis techniques which we also exploited to analyze the distribution of sentiment in the different communities found. In this regard, in the future an emotion analysis could also be performed to obtain even more detailed information on user feedback.

As for topic modeling, we observe from the word clouds and our interpretation that the topics primarily revolve around opinions on the movie's quality and plot, or considerations about the story and characters. Therefore, despite the film being among the highest-grossing in cinema history [10], none of the seven detected topics seems to reference the box office success or the film's production.
Topic modeling could be further developed, considering that, as mentioned earlier, the choice to proceed with 7 topics is justified by a good balance between the values of the two parameters, but it is not the only viable option. Therefore, it might be worthwhile to run the model with a different number of topics, for instance 4 or 5, and assess the results.
Furthermore, one could explore the use of alternative models to LDA, such as LSA (Latent Semantic Analysis) or BERT (Bidirectional Encoder Representations from Transformers). In these cases, text representation strategies other than BOW, such as TF-IDF, could be considered. Implementing these models with a range of possible topics from three to nine would allow for a comparison of `U_mass` and `C_v` coherence values with those obtained with our LDA model.

References

[1] Reddit's best communities:
https://www.reddit.com/best/communities/1/

[2] GitHub - PRAW:
https://github.com/praw-dev/praw

[3] pycld2 library:
https://pypi.org/project/pycld2/

[4] Gephi:
https://gephi.org/features/

[5] Louvain algorithm:
Towards data science - Louvain Algorithm. Luìs Rita, 09/04/2020

[6] VADER:
https://pypi.org/project/vaderSentiment/

[7] Gensim:
https://pypi.org/project/gensim/

[8] Ismail Harrando, Pasquale Lisena and Raphael Troncy.Apples to Apples: A Systematic Evaluation of Topic Models:
https://aclanthology.org/2021.ranlp-1.55.pdf

[9] pyLDAvis:
https://pypi.org/project/pyLDAvis/

[10] movieplayer.it, film con maggiori incassi nella storia del cinema:
https://movieplayer.it/film/boxoffice/internazionale/di-sempre/