Topic Modeling of ChatGPT users on Reddit

Claudio De Martino
c.demartino1@studenti.unipi.it
Student ID: 560934

Giuliano Galloppi
g.galloppi@studenti.unipi.it
Student ID: 646443

Clara Lavorati
c.lavorati@studenti.unipi.it
Student ID: 565091

Isabel Santucci
i.santucci@studenti.unipi.it
Student ID: 597351

ABSTRACT

Nowadays, ChatGPT is on everyone's lips. Based on an LLM that uses large neural networks to learn and generate text, it is able to answer questions, provide information, perform specific tasks and participate in conversations. The aim of this project's analysis, using Network Science tools, is to investigate how discussion around ChatGPT takes place among users of the social network Reddit, and which topics these fall under, decreting a majority use in a particular area exploiting some interesting statistics from it. [1]

KEYWORDS

ChatGPT, AI, Reddit, Topic Modeling, LLM, Social Network Analysis

ACM Reference Format:
Claudio De Martino, Giuliano Galloppi, Clara Lavorati, and Isabel Santucci. 2023. Topic Modeling of ChatGPT users on Reddit. In *Social Network Analysis '23.* ACM, New York, NY, USA, 12 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

ChatGPT is an artificial intelligence model developed by OpenAI, also called Generative Pre-trained Transformer 3 (GPT-3), designed to understand and generate natural text, enabling interaction with it through written conversations.

Transformers are deep learning models that have revolutionised the field of Natural Language Processing (NLP). Introduced in 2017 by Vaswani et al.[10], they rely on the 'attention' mechanism to handle sequences of words efficiently and capture long-range relationships. This technology has led to significant improvements in the performance of NLP models.

It is used nowadays in any field and by anyone as a virtual assistant who can answer your questions, provide any kind of information and even support a conversation. The massive use of chatbots is increasingly evolving into the main subject of discussion today: this is the reason that prompted the group to analyse in more detail the discussion that develops around these tools using the social network 'Reddit' [1] as the main data source.

The purpose of this analysis is to investigate the interactions between users, which take place through the exchange of mutual comments, using the knowledge and techniques of Social Network Analysis in order to be able to answer the research question that arose spontaneously to identify in which areas ChatGPT is most discussed; this question was answered by performing a Topic Modeling analysis on the users belonging to the collected data, dated December 2022. The development process of the project initially consisted of a Data Crawling phase from the Social Network, followed by a Pre-processing of the data to make it adaptable to our analysis, after constructing a network graph made of nodes (users) and edges (comments between users). The main metrics that were employed to elaborate the Network Analysis phase include: degree distribution, connected components, path analysis, clustering coefficient, density analysis, centrality analysis. These were then compared with ER and BA graphs constructed with equal numbers of nodes. In the Data Analysis and Network Manipulation phase instead, Community Detection, Spreading, Dynamic Network Analysis and Higher-order Network Analysis tasks were explored, which are detailed in the following sections. Finally, an analysis of the open question discussed earlier on topic modeling with the obtained results is given.

2 DATA COLLECTION

To achieve the goal of this project, we decided to retrieve the data from the Social Network Reddit.[1]

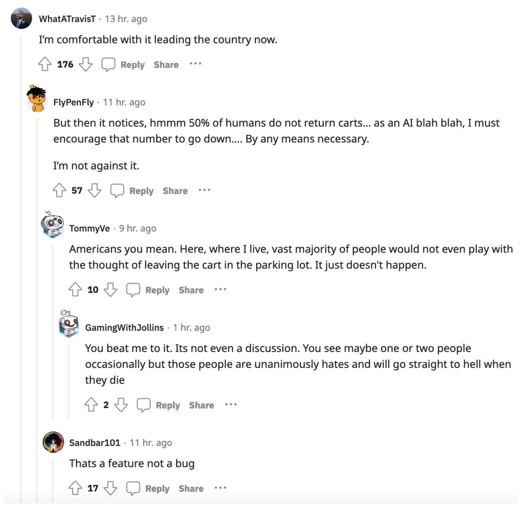
[1]**Project Repository:**
https://github.com/sna-unipi/sna-2023-2023_de-mart_gall_lavo_sant

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.
SNA '23, 2022/23, University of Pisa, Italy
© 2023 Copyright held by the owner/author(s). Publication rights licensed to ACM.
ACM ISBN 978-x-xxxx-xxxx-x/YY/MM…$0.00
https://doi.org/10.1145/nnnnnnn.nnnnnnn

Reddit is a content sharing Social Network that allows users to discuss and share information on a wide range of topics. It is based on a system of sub-networks called ‘subreddits’: every subreddit is dedicated to a specific topic to which users can subscribe in order to share content, join discussions and express their preferences using a voting system on posts and comments through the ‘upvote/downvote’ buttons. The algebraic sum of the upvotes and downvotes determines the score of the post or comment.

Selected Data Sources

From a first direct scraping from Reddit using the official APIs through wrappers such as Praw, Pmaw and Psaw, we found an inadequacy of the available scraping methods or the deactivation of the last two. Thus, we decided to use official data backup of the social network dating back to December 2022 to avoid inconsistencies of the most recent data. The data was collected from r/ChatGPT subreddit.



WhatATravisT · 13 hr. ago
I'm comfortable with it leading the country now.
176 Reply Share ···

FlyPenFly · 11 hr. ago
But then it notices, hmmm 50% of humans do not return carts... as an AI blah blah, I must encourage that number to go down.... By any means necessary.

I'm not against it.
57 Reply Share ···

TommyVe · 9 hr. ago
Americans you mean. Here, where I live, vast majority of people would not even play with the thought of leaving the cart in the parking lot. It just doesn't happen.
10 Reply Share ···

GamingWithJollins · 1 hr. ago
You beat me to it. Its not even a discussion. You see maybe one or two people occasionally but those people are unanimously hates and will go straight to hell when they die
2 Reply Share ···

Sandbar101 · 11 hr. ago
Thats a feature not a bug
17 Reply Share ···

Figure 1: Exchange of comments in a Reddit post

Crawling Methodology and Assumptions. The dataset called ‘reddit_comments.csv’ contains information about Reddit users and their comments w.r.t. the replies among them, as we can see in the Figure 1.

It contains 49886 records, which go from 2 December 2022 to 31 December 2022, and 10 attributes.
The most relevant of them are:

- *date*: comment creation date in DD/MM/YYYY format;
- *clean_text*: comment's text;
- *author*: unique username of the comment author;
- *parent_author*: unique username of the user who has been replied to by commenting (added later);
- *created_utc*: date the scraped record was added, in UTC format;
- *num_comments*: number of comments, valid only for posts, NaN otherwise;
- *score*: comment score obtained by total number of upvotes/downvotes, popularity and discussion engagement;
- *type*: type of data collected, post or comment.

The collected data is guided by the assumption the group made for the construction of the graph, i.e. the *nodes* are represented by the users in the dataset, while the *edges* are represented by the exchange of comments between users (i.e. if a user responds to a comment of another user then there is a relationship between them). The decision was made to represent the graph in an ‘oriented’ manner, motivated by the structure of the replies between users; it returned 13171 nodes, which are the users, and 40948 edges, which are the comment exchanges.

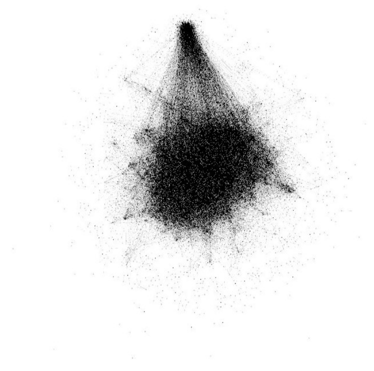


Figure 2: Representation of the Reddit network in Gephi

From these dataset fields we created two .csv files containing respectively only the node and edge data, useful for their reuse in each project task, in order to build the network. We used the undirected version of the graph to simplify the analysis.

3 NETWORK CHARACTERIZATION

The following analyses were performed on the entire set of nodes. For this part we mainly relied on the networkx [7] library.

Degree distribution analysis. First of all we calculated the degree of each node and the average degree of the network. The approximate result is 5.21. We have also developed three graphs on the degree distribution that allow a more complete view.

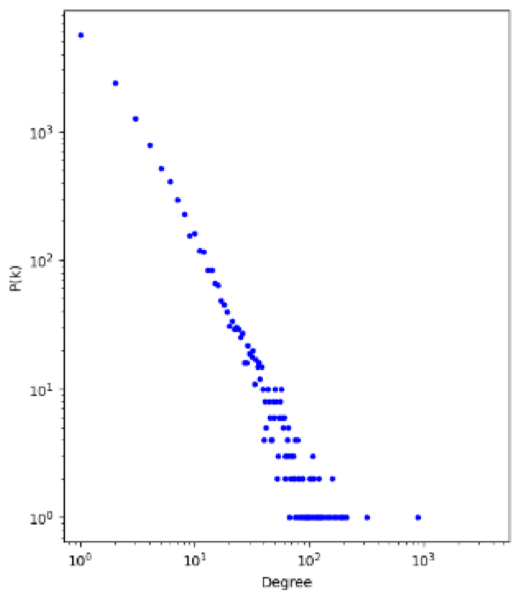


Figure 3: Degree distribution

In Figure 3, the graph simply plots the degree values on the X axis and the number of nodes for each degree value on the Y axis.

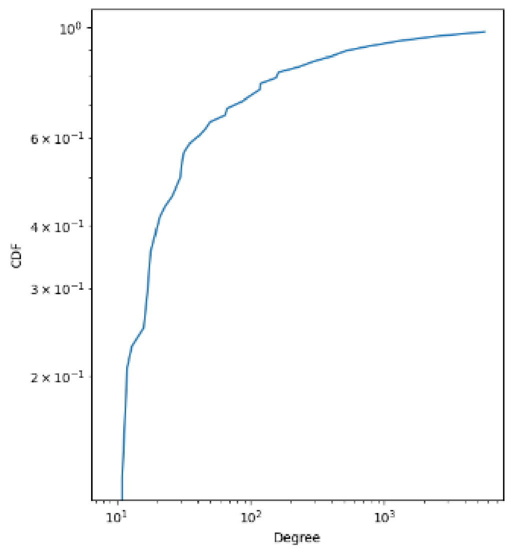


Figure 4: CDF (Cumulative Distribution Function)

The graphs in Figure 4 and Figure 5 represent respectively the Cumulative Distribution Function and the Tail distribution. The trend of both graphs shows that our network has an exponential distribution.

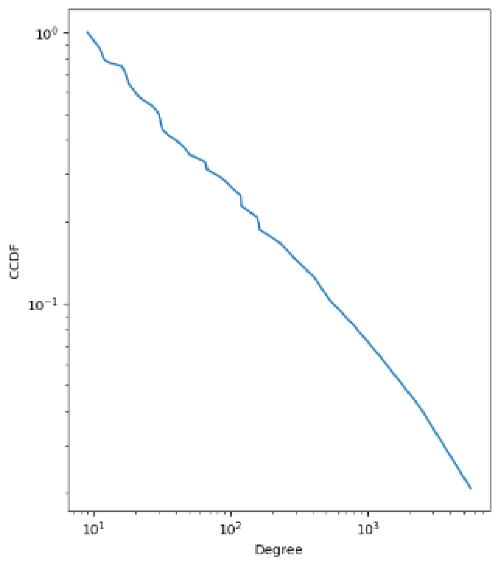


Figure 5: CCDF (Complementary Cumulative Distribution Function)

Connected component analysis. Before calculating the number of network components we wanted to try to represent the ego-network of the first node but the graph was not interesting. Using the *connected_components()* method we calculated the number of components, which turned out to be 299. We then tried to identify a giant component, removing all nodes that were not part of it. The result is that we have gone from 299 to a single connected component and this component is made up of 12676 nodes. If the initial number of nodes was 13171, it means that the other 298 connected components were made up of 495 nodes, i.e. less than 4% of the total number of nodes.

Path analysis. The diameter measures 12 and the average shortest path corresponds to 4.31. We can notice that the average shortest path is 1/3 of the diameter.

Global clustering coefficient and Density. The coefficient is equal to 0.045, which is a low value, meaning that the nodes on the graph have a low tendency to form closely related groups or clusters and there is an high number of weak ties. Additionally, the density is 0.00042: it suggests the graph is relatively sparse, since the density indicates the fraction of connections actually present compared to the maximum number of connections possible. In this case there are few connections between nodes.

Centrality analysis. Firstly, we investigated the number of interactions of each node by computing their degree centrality. The following table shows the top five nodes with the highest degree.

Table 1: Degree Centrality

Username	Valore
hi_there_bitch	892
TerrySharpHY	321
TwoCoresOneThread	212
perturbaitor	199
ClinicalIllusionist	195

Then we analysed the centrality of the nodes in the network from different points of view using the following measures: closeness centrality, betweenness centrality, harmonic centrality, PageRank and EigenVector centrality.

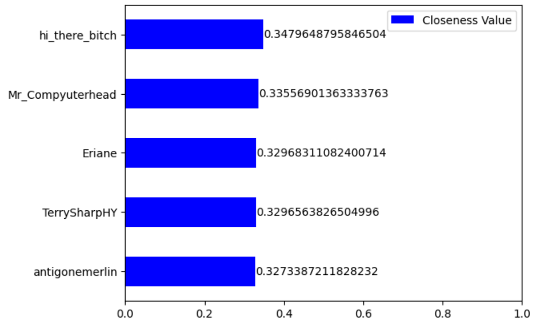


Figure 6: Closeness Centrality

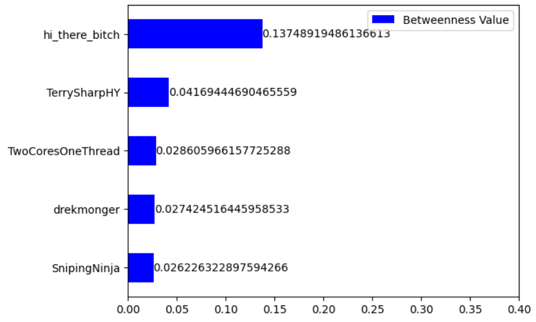


Figure 7: Betweenness Centrality

Figures 6 and 7 show the results of the closeness and the betweenness centrality analysis. Closeness centrality is a measure that quantifies how influential a node is in a graph and it enables us to evaluate how easily a node can reach other nodes in the graph; while betweenness centrality assesses the importance of a node based on its ability to act as an intermediary between other nodes. By looking at the results shown in Figure 7, one could argue that there are no nodes that play a critical role in connecting different parts of the network. In fact, the betweenness centrality scores are very low, below 0.15. It can also indicate that the network is made of unconnected components.

Table 2: Other Centrality Measures

Eigenvector		PageRank		Harmonic	
Username	Value	Username	Value	Username	Value
hi_there_bitch	0.505	hi_there_bitch	0.015	hi_there_bitch	5182.211
TerrySharpHY	0.116	TerrySharpHY	0.005	Mr_Compyuterhead	4758.623
antigonemerlin	0.108	TwoCoresOneThread	0.003	TerrySharpHY	4721.580
Mr_Compyuterhead	0.107	ClinicalIllusionist	0.002	Eriane	4656.199
perturbaitor	0.101	perturbaitor	0.002	antigonemerlin	4636.908

As we have seen so far, the users that play the most central roles are basically the same in every centrality analysis we conducted and they are also the ones with the highest number of connections. Finally, we employed Newman's Assortativity to quantify homophily, namely the tendency of nodes with similar characteristics to be connected to each other. The resulting assortativity coefficient is: -0.066. A negative assortativity coefficient indicates disassortative mixing, thus nodes tend to connect to others with different properties. For instance, high-degree nodes connect with low-degree nodes.

Comparision with ER and BA graphs

To build the Erdos-Renyi graph (ER) we needed the value of the probability of the edges creation. In order to find it, we used the following formula:

$$p = \frac{2L}{N(N-1)} = \frac{240948}{13171 \cdot 13170} = 0.0004721262694$$

Since the ER graph is constructed through the use of a probability, the number of edges resulting at each execution of the code varies, although they remain around the real value. While for the construction of the Barabási-Albert graph we calculated the number of edges by the number of nodes multiplied by m, which we obtained from the following formula:

$$m = \frac{\langle k \rangle}{2} = \frac{6.217902968643231}{2} = 3.1089514843216155$$

Thus, the ER graph has 40739 edges, while the BA graph has 39504 edges.

Comparing the results of the degree distribution with those of the real network we noticed that the average degree in the ER graph changes a little, approximately 6.18, compared to 6.21, while the number of connected components drops to 28, compared to the 299 of the real network. Even the graphs related to the degree distribution change a lot, they no longer represent an exponential distribution, but instead they show a normal one (Figure 8, 9 and 10). The regime type of this graph is supercritical, as the value of p is greater than 1/N, which is 0.000076. The BA graph, on the other hand, is closer to the results of the real network, with the average degree being approximately 6 (5.99) and also the degree distribution graphs are similar to those of the real network. The number of connected components is 1, resulting in a single giant component.

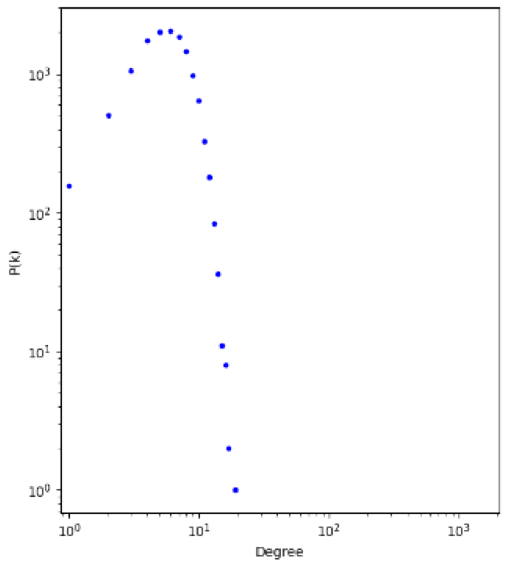


Figure 8: Degree distribution of ER graph

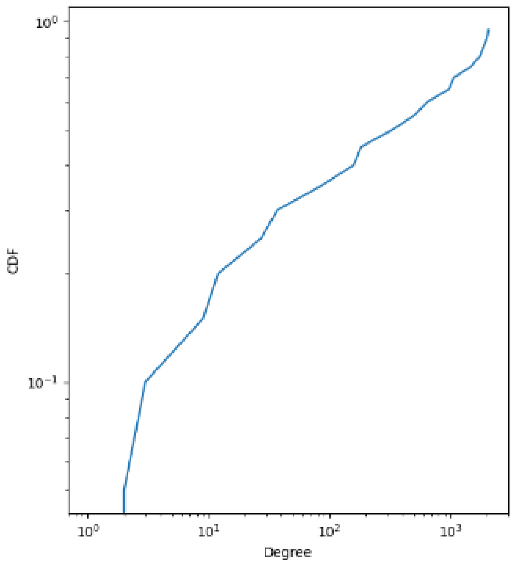


Figure 9: CDF of ER graph

With regard to the centrality analysis, for both Erdos-Renyi and Barabási-Albert graph we have chosen to display the top five results of the following centrality measures: Degree, Closeness and Betweenness centrality (Table 3). Nevertheless, the Degree Centrality results appear to be the most relevant.

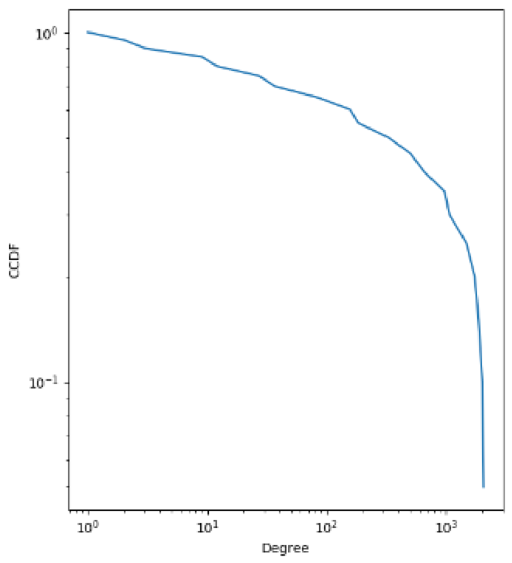


Figure 10: CCDF of ER graph

Table 3: Centrality Measures in ER and BA graphs

ER GRAPH			BA GRAPH		
Degree C.	Closeness C.	Betweenness C.	Degree C.	Closeness C.	Betweenness C.
21	0.214	0.002	351	0.35	0.098
18	0.213	0.002	291	0.35	0.083
18	0.21	0.002	287	0.34	0.069
17	0.21	0.001	228	0.34	0.062
17	0.21	0.001	180	0.34	0.047

In terms of nodes degree, the difference between the random network predictions and the real world network is attributable to the Poisson distribution, which minimizes the possibility of a node to have a high degree. In fact, according to the ER model the maximum degree should be 21.

4 NETWORK MANIPULATION AND ANALYSIS

Community detection

This analysis focuses on discovering the network's structure, comparing and evaluating the results of the identified communities, using Cdlib [6] library. Concerning crisp communities, the algorithms used are Louvain and Leiden. Louvain algorithm has two phases repeated iteratively. Each node begins in its own community, then moves to the adjacent community that offers the biggest modularity boost. The graph is updated with communities connected by weighted links and bridges. Leiden algorithm works as a hierarchical clustering that merges communities into single nodes.

As far as overlapping communities are concerned, namely a node can belong to multiple communities, two algorithms are implemented: Demon and k-clique. Demon algorithm starts from the ego-networks, built upon a focal node. The ego node n is extracted and removed from the network, then label propagation is performed and n is placed in every identified community. Finally, if some communities are similar they are merged. Instead, a k-clique community is the union of all k-size clicks that can be achieved by adjacent k-cliques (sharing k-1 nodes).

Community-based evaluation is implemented through two strategies: internal evaluation and external evaluation.

Internal Evaluation

- **Conductance**: the fraction of total edge volume that points outside the community. Particularly, lower values are better than the higher ones.

$$f(S) = \frac{cS}{2mS + cS}$$

In the formula *cs* stands for number of community nodes and *ms* represents the number of community edges

- **Internal edge density**: the proportion of edges connecting nodes within a specific community to the total number of possible edges within that community. Specifically, lower values are better.

$$f(S) = \frac{mS}{\frac{nS(nS-1)}{2}}$$

where *mS* is the number of community internal edges and *nS* is the number of community nodes.

External Evaluation

- **Overlapping Normalized Mutual Information LFK** between two clusterings. It's an extension of the Normalized Mutual Information (NMI) score to cope with overlapping partitions.

$$ONMI = \frac{MI(P,R)}{\sqrt{H(P) \cdot H(R)}}$$

where *MI(P, R)* represents the mutual information between the partition P and the reference labels R. *H(P)* is the entropy of the partition P and *H(R)* is the entropy of the R reference labels. The value obtained is between 0 and 1, where 1 indicates a perfect match between the overlapping partition and the reference labeling, while values closer to 0 indicate less similarity.

- **F1 score**: used to measure the quality of partitions obtained from the community detection algorithm.

$$F1 = \frac{2 \cdot (precision \cdot recall)}{(precision + recall)}$$

where *precision* measures the proportion of correctly detected items to total detected items and *recall* indicates the proportion of correctly detected positive items to the total number of positive items present.

Table 4: Internal evaluation results

	Parameter(s)	#Community	Conductance	Internal edge density
Louvain	Resolution:2	383	0.12	0.46
Leiden	None	335	0.047	0.53
Demon	ϵ:0.25	110	0.85	0.11
k-clique	K:3	602	0.87	1.18

Concerning conductance, the best values are represented by Louvain and Leiden algorithms, so they show an internal cohesion respect to Demon and K-clique. For internal edge density the ones who performed better are K-clique and Leiden, indicating that nodes within the community are fully connected to each other.

Table 5: External evaluation results

Partitions	ONMI LK	F1 score
Louvain and Leiden	0.83	0.81
Demon and k-clique	0.05	0.07

In Table 5 we can observe that the best results are related to Louvain and Leiden partitions, which reveal almost complete match regarding overlapping partition. While the F1 score indicate that they have a better quality community partition, i.e. nodes have been classified with greater accuracy, than Demon and k-clique.

Another way to compare the different communities, taking into account a particular evaluation measure, can be obtained using an heatmap (Figure 11).

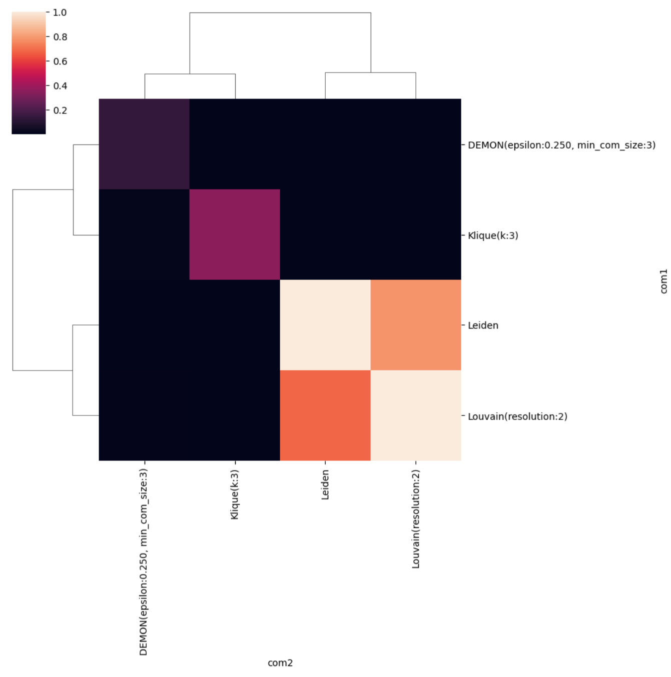


Figure 11: Heatmap of different communities by nf1 evaluation

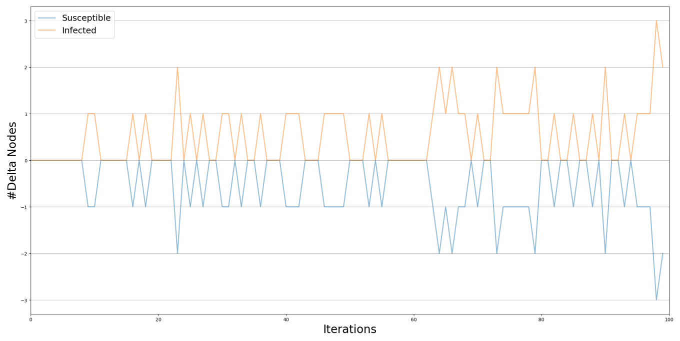


Figure 12: SI model

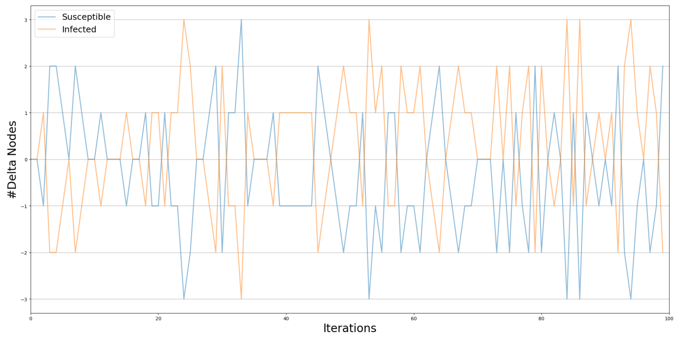


Figure 13: SIS model

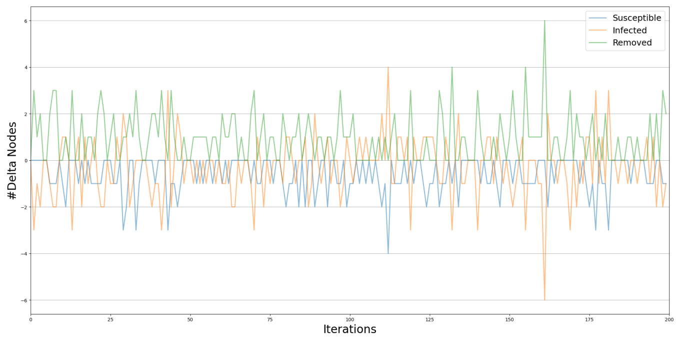


Figure 14: SIR model

Spreading

Diffusive phenomena are analyzed using the *ndlib* [9] library by considering the three stochastic epidemic models: SI, SIS, and SIR. In the first one the individuals can be in one of these two states: susceptible (S) or infected(i).

In SIS model susceptible individuals become infected and each has a probability to revert its status to susceptible; whereas SIR susceptible individuals got infected but each one has a probability of becoming immune. These models applied to the network show symmetric pattern, so when a node becomes infected, its connectivity with other nodes might allow a high probability of transmission, causing an increase in infected nodes. Later, when the number of infected nodes reaches its peak, the effect of healing or control measures could cause a decrease in the number of infected nodes, leading to a symmetrical trend in the graph. Although this distribution can be almost identical for every network, there may be differences in values due to system characteristics. Significance can be gained from analyzing the SIS and SIR models utilized, and some remarks can be made. The first one (Figure 13) has 6 peaks at different iterations with the delta value (3, -3). The SIR model (Figure 14) only has one peak approximately at the 160th iteration with a delta value between 6 and -6.

Higher-order Network Analysis

Formally, a hypergraph is defined as a pair H = (V, E), where V is a set of nodes and E is a set of hyperedges. Each hyperedge in E can contain any number of nodes. To build the Hypergraph we used the library *HyperNetX*[8]. We adapted our dataset to the format the library requested: we created a new attribute that specifies the edges between two nodes through a label. The resulting number of hyperedges is 49885. Then we investigated the cardinality of the hyperedges, which is the number of nodes contained in a hyperedge.

Table 6: Hyperedge Information

Cardinality	Number of Hyperedges
2	48096
1	1789

Hyperedges with two nodes are equivalent to traditional edges in a graph. The fact that the majority of hyperedges have a cardinality of 2 implies that the hypergraph focuses on pairwise connections; this also suggests the absence of higher-order relationships and complex interactions between larger sets of nodes. The hyperedges with cardinality equal to 1 represent a single node connected to itself: it could indicate the presence of self loops or isolated nodes. A visual representation of the hyperedges is portrayed in Figure15, in which hyperedges of both cardinality 1 and 2 are rendered.

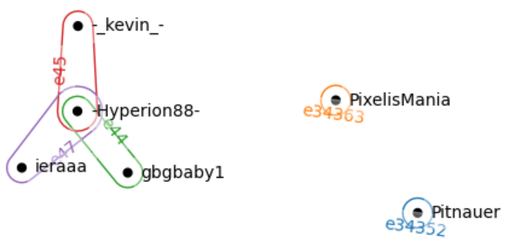


Figure 15: Hyperedges.

This situation clearly represents the direct interactions between pairs of users in the network.
Additionaly, we devided the number of hyperedges by the number of nodes to obtain the connectivity measure. The result was 3.79.
Finally, we investigated the modularity of the hypergraph to measure the quality of its partions. Before applying the modularity algorithm we used Kumar clustering algorithm, which is a method that uses the Louvain algorithm to create a partition of the vertices. The resulting modularity score is 0.51. The modularity score ranges from -1 to 1, with higher values indicating a stronger community structure. Thus, this modularity score suggests a moderate level of community structure in the network.

Dynamic Network Analysis

The Dynamic Network Analysis was done using *DyNetX*[5], a Python library that extends networkx for dynamic network modelling.
We modelled our data by creating several 'Network Snapshots', then generating Snapshot Graphs to represent, as a dynamic phenomenon to be observed, the evolution of our graph over certain time intervals.
In the phase preceding the construction of the Dynamic Network, of which the Snapshots Graph is composed, the time interval was discussed.

Data pre-processing / DynGraph settings. The dataset on which our analysis is based contains data from December 2022, therefore the creation of the Dynamic Network focused on a 4-week time interval, observing its evolution with each addition of the arcs according to their intuitively increasing periods of appearance.

The total dates on which we investigate are 30, from 02/12/22 to 30/12/22 inclusive. Having imported a time window of 7 days, the total number of groups/classes indicating them, (i.e. interactions/edges) is 4. Therefore, each row of the dataset has been labelled by adding a *Time t* column according to the corresponding date, in ascending date order.

The Dynamic Graph is then constructed by adding the edges from the dataset in order of time of appearance, so from time 0 to time 4, while the nodes of the network exist from time 0, resulting with 13170 nodes and 33741 edges.

Snapshots Analysis. From the Dynamic Graph we can access to each different temporal snapshot, whose data are reported in the following Table 7.

Table 7: Snapshot's Nodes and Edges

Snapshot	Number of Nodes	Number of Edges
0	1798	3186
1	6250	13449
2	9838	23317
3	13170	33741

From the number of nodes and edges, we can see that as the time intervals grow, and therefore as edges are added to the network, we can say, as we anticipated earlier, that the network grows. In fact, we note that in the last snapshot, where we have the complete graph showing the data for the entire month, it includes all nodes and edges.
Similarly, it is possible to obtain snapshots of longer time intervals, e.g. we can obtain data from a Snapshot ranging from time t=0 to t=2 and see the state of the graph in that time span:

```
Number of nodes in the interval: 9838
Number of edges in the interval: 23317
Interaction per Snapshot 0: 1631.0, 1: 6855.5
```

We note that nodes and edges correspond precisely to those in Snapshot 2 in the Table 7 above. While for the interactions, the 0 key indicates interactions observed in the first week of the month, instead the 1 key indicates interactions observed from the first to the second week, i.e. it counts whether the arcs starting from the observed week remain present until the end of the time interval.

Dynamic Network Measures. Finally, in the next tables we can observe the measurements performed on the constructed Dynamic Graph, some of which refer to two generic nodes u and v, chosen randomly.
The first measures, reported in Table 8, regard the Inter Event Time Measure (IET), that indicates how much time occurs before a new interaction appears in the graph.

Table 8: Inter Event Time Measures

IET	N. Interactions
Global	temporal distance {0: 34728, 1: 3}
Node	temporal distance {0: 2, 2: 1}
Edge	temporal distance {3: 1}

We can notice that in the value of Global measurement, therefore taking into account the whole month, just in one of the four snapshots there are three new interactions.

In Table 9 below, instead, are reported the measures of: *Coverage*(C) that is the ratio of existing nodes w.r.t. the possible ones; *Coverage of Node/Edge* (CNE) which reports the coverage of w.r.t. a node 'u' and of edge (u, v), *Pair Uniformity* (PU) that is the overlap between the presence times of u and v; *Density* (D) that indicates the temporal network density, which is the fraction of possible interactions that do exist in the temporal network and *Node Density* (ND), *Pair Density* (PD) which respectively indicate the intersection between the node/edge's temporal presence and the joint node/edge's temporal presences.

Measure	Value
C	0.59
CNE	{Node: 1.0, Edge: 1.0}
PU	1.0
D	0.0004634
ND	0.0004507
PD	1.0

Table 9: Dynamic Network Measures

The *Snapshot Density* measure, on the other hand, shows us that the Snapshot Networks taken individually all give very low values, with the densest being the one at snapshot one with a value of 0.0020, consistent with the observed density of a single node w.r.t. the entire graph in Table 9; as the network grows, it decreases more and more, emphasizing the increase in edges but the sparsity of the inspected network.

Study of stability. The stability study was done to observe changes over time between snapshots by questioning four metrics: Jaccard Coefficient, Clustering Coefficient, Degree Centrality, and a comparison of the Degree Distribution.
The *Jaccard Coefficient*: the highest value was recorded between Snapshots at time t=2 and t=3 with 0.67 which tells us that there were few changes in that timeframe, i.e. there was a low addition to the exchange of responses; whereas the lowest value was recorded between Snapshots at time t=0 and t=1 with 0.22, which tells us that there were few nodes that interacted with each other between the first and second week.
The *Clustering Coefficient*: The highest value was recorded in the Snapshot at time t=0 with 0.057 which tends to decrease slightly until the last Snapshot at time t=3 with 0.043. They both remain very small values, which indicate that the tendency of a node's neighbours to be connected to each other is very low from the start, and as the network grows this tends to decrease.
The *Degree Centrality*: as the results we obtained show, the node that has the highest degree centrality in every snapshot is *'hi_there_bitch'*, which has the value of 0.11 at time t = 2. Despite this fact, the values of degree centrality of all nodes prove that the network is sparse at all times.

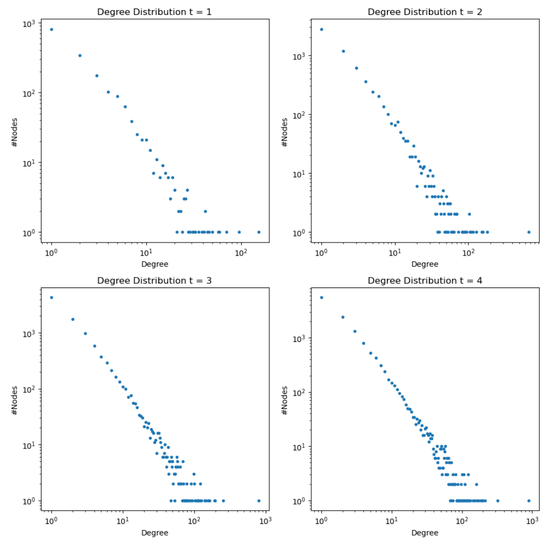


Figure 16: Degree Distribution of each Snapshot

In conclusion, as we can see in the Figure 16 there are few changes in density, taking into account a total time window of one month, indicating again that the network is evolving over time, growing slowly.

5 OPEN QUESTIONS

In this section we investigate which are the main subject areas the comments fall into. The research question we asked ourselves was: *'In which topics is chatGPT most used/discussed by users on Reddit?'*. In order to determine that, we carried out a topic modeling analysis. We used a transformer-based model named BERTopic [2], which is an open-source library that uses a BERT model to do Topic Detection with an TF-IDF (Term Frequency - Inverse Document Frequency) algorithm that weights the importance of words in a corpus. The importance depends on the word's frequency: the more frequent a word is, the more important it is. The aim of the analysis is to get an overview of the main topics of the comments we retrieved.

We prepared the data with a pre-processing that included the removal of the stop words, since their high frequency can cheat the extraction. We also looked for capital letters and special characters. After this verification, the list of the comments was submitted to the model. Only a small representative portion of 1300 comments was given.
The topics extracted were visualized through different kinds of graphs. The most relevant is the barchart (Figure 17) and it shows the main topics identified and for each one it displays the internal subjects covered.

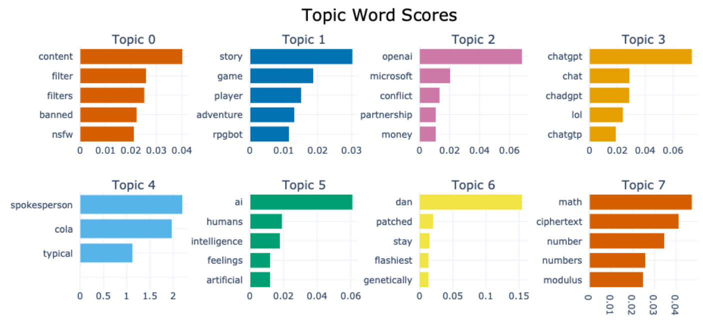


Figure 17: Barchart of the Topic Word Scores

Finally, the test part is performed through the use of different probability measures that reveal to which extent the sentences belong to the topic identified. There are three parameters:

- **Topic**: which contains the id of the topic
- **Count**: that refers to the number of the sentences belonging to a particular topic
- **Name**: the automatically generated topic label.

The top '-1' topic is assumed as irrelevant.

The analysis provided eight different topics. We found the most relevant to be: *history, games, Microsoft* and *mathematics.*

After the recognition of the main topics, other specific analyses are carried out, namely sentiment analysis and the detection of the dominant matter in each topic.

Insight into the topics

After the topic identification by Bert, an in-depth study about them is developed to understand more specifically what people could ask to ChatGPT in relation to the different arguments identified. The analysis starts from the Named-Entity Recognition (NER) [4], a method that extracts information from the given text. It consists of identifying the named entities referring to the key topics of the text, in this case comments, such as names, locations, companies, products, events. First, the comments about that topic are extracted by using topic keywords such as *history*, *game*, *Microsoft*, *math*. The NER process is accomplished by considering only PER, which is the name of people, and ORG, which is the name of different types of organizations. Frequency of words is achieved by obtaining key words that can inform about the most commonly discussed topic among people. Lastly, a graphic linking the keywords, represented by nodes, is constructed. The number of words is not a fixed number, as everything is dependent on the topic.

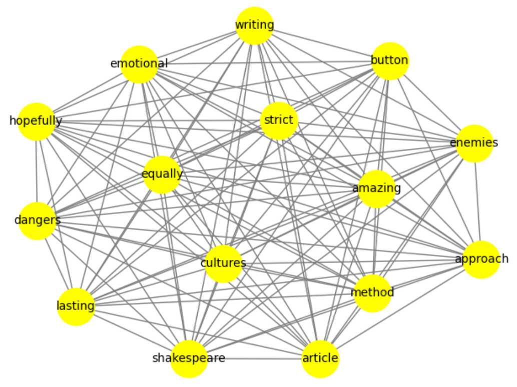


Figure 18: Graph for history topic

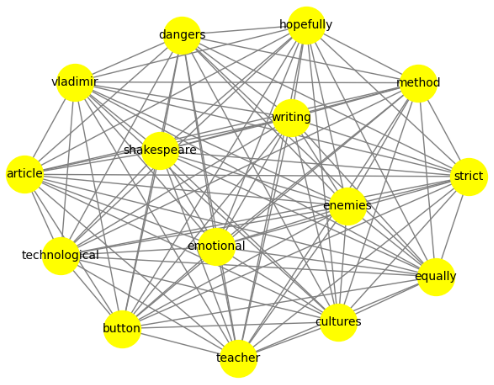


Figure 19: Graph for game topic

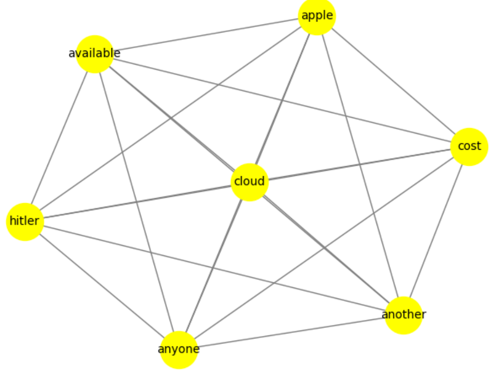


Figure 20: Graph for Microsoft topic

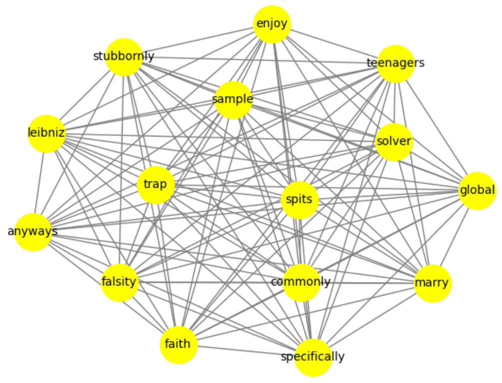


Figure 21: Graph for math topic

In the following results what emerges are words related to topics. The first graph (Figure 18) about history contains terms such as *Shakespeare*, *enemies*, *method*, *writing*, *cultures*. It could be possible that people asked ChatGPT to help them with a task, for instance writing an article or looking for one.

In the second one (Figure 19), that concerns the gaming area, there are several common words such as *dangers*, *method*, *technological*, *enemies*. Probably users talk about games to ChatGPT to find some strategies or just to take some information.

The third graph (Figure 20) is related to Microsoft, so what is expected is to find some technological terms, indeed there are words as *cloud*, *avalaible*, *apple*, but what is stranger is to find terms like *hitler*, that should belong to the history topic. In the math topic graph (Figure 21) there are few words connected to the argument as *Leibniz*, *solver*, *sample*, *specifically*, but the majority of the words are not relevant. These last two graphs are the ones that show the highest amount of noise; nevertheless, it is possible to infer some information about the area they cover.

Sentiment analysis. Through this analysis the general idea of the users' opinion about ChatGPT can be interpreted. The library used is *TextBlob* [3], which is a Python library for natural language processing (NLP) that simplifies information extraction from text and offers features such as sentiment analysis, sentence extraction, tokenization, lemmatization, part-of-speech classification, and more. For each sentence the *sentiment* is computed and it is delivered as a score; in relation to it the sentence can be classified as Positive (score greater than 0), Negative (score less than 0), or Neutral (other score values). The visual distribution is realized by a barchart (Figure 22) , which discloses that the majority of the opinions about ChatGPT are rated as positive.

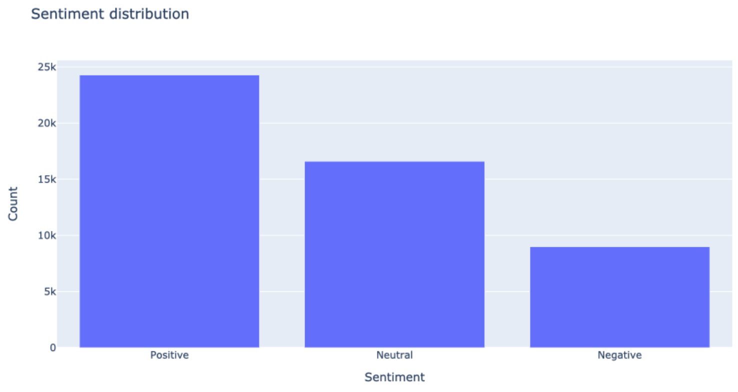


Figure 22: Sentiment analysis about Chat GPT

6 DISCUSSION

In this project, we have investigated the interactions among the exchange of comments of Reddit users, with the final goal to answer the research question of: 'In which topics is ChatGPT most discussed?'. The data collection phase was carried out by drawing from a backup of the ChatGPT subreddit dated December 2022. The analysis of the network structure shows the network is sparse because of the low clustering coefficient, low density and the small average shortest path. An analysis of the dynamic version of the network was carried out: we noticed that the number of edges constantly increases over the weeks and that between the time slices analysed there is coherence underlining the evolution of the network linked to its sparsity. Lastly, using BERTopic to do a topic modeling analysis, we found the main topics that the comments belong to. The ones that gave the best results were history and gaming.

REFERENCES

[1] 2005. Reddit. https://www.reddit.com/

[2] 2018. BERTopic. (2018). https://www.kaggle.com/code/meetnagadia/topic-modeling-using-bertopic

[3] 2021. TextBlob. (2021). https://textblob.readthedocs.io/en/dev/

[4] 2023. Named-Entity Recognition by Spacy library. (2023). https://www.analyticsvidhya.com/blog/2021/06/nlp-application-named-entity-recognition-ner-in-python-with-spacy/

[5] Giulio Rossetti et al. 2017. DyNetx. (2017). https://github.com/GiulioRossetti/dynetx/tree/master

[6] Giulio Rossetti et al. 2023. cdlib. (2023). https://cdlib.readthedocs.io/en/latest/

[7] Aric Hagberg, Pieter Swart, and Daniel S Chult. 2008. Exploring network structure, dynamics, and function using NetworkX. Technical Report. Los Alamos National Lab.(LANL), Los Alamos, NM (United States).

[8] Brenda Praggastis, Dustin Arendt, Cliff Joslyn, Emilie Purvine, Sinan Aksoy, and Kyle Monson. [n. d.]. pnnl/HyperNetX.

[9] Giulio Rossetti, Letizia Milli, Salvatore Rinzivillo, Alina Sîrbu, Dino Pedreschi, and Fosca Giannotti. 2017. NDlib: a python library to model and analyze diffusion processes over complex networks. International Journal of Data Science and Analytics 5, 1 (dec 2017), 61–79. https://doi.org/10.1007/s41060-017-0086-6

[10] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention Is All You Need. arXiv:1706.03762 [cs.CL]