# Final presentation NS project

Filippo Canderle
Cecilia Rossi
Anna Zorzetto

20 febbraio 2024

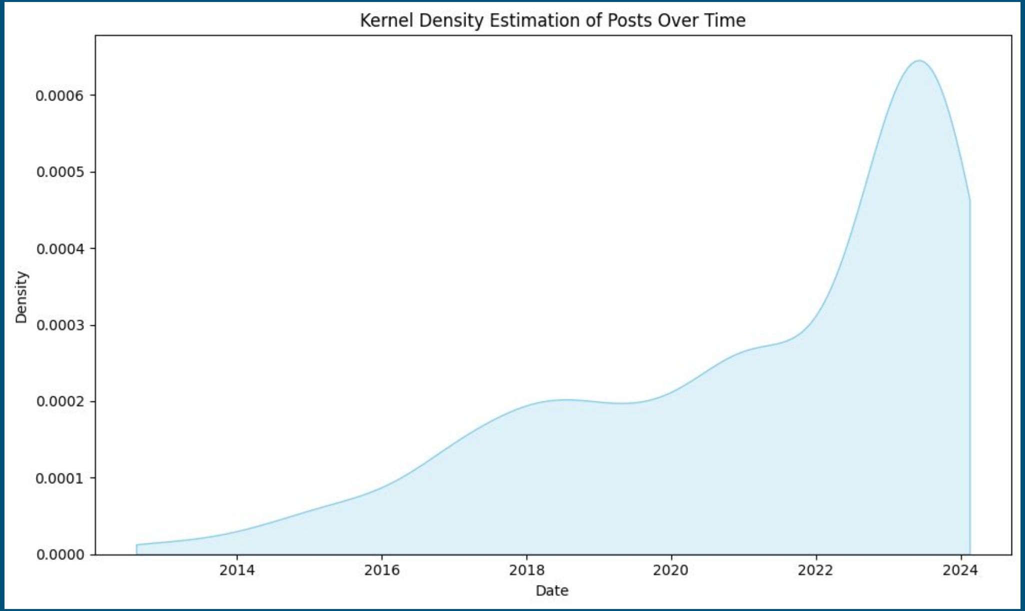
# Introduction

The aim of our project is the comparison of two semantic networks:

pre chat GPT and post chat GPT

# Data Searching

We looked for 100 subreddits with keywords ‘Artificial intelligence OR AI or IA’ and we analyzed 2000 posts within each subreddit



# Data Cleaning

We performed the cleaning in two steps:

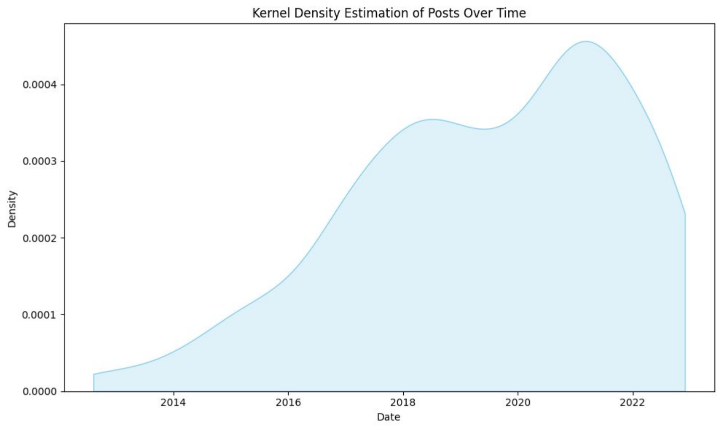
1. superficial cleaning: to polish the text from contractions, emoji, HTML tags, special characters, numbers, whitespaces, links to websites
2. deep cleaning: to add part of speech to each word according to POS_KEEP = ['ADJ','ADV','NOUN','PROPN','VERB']

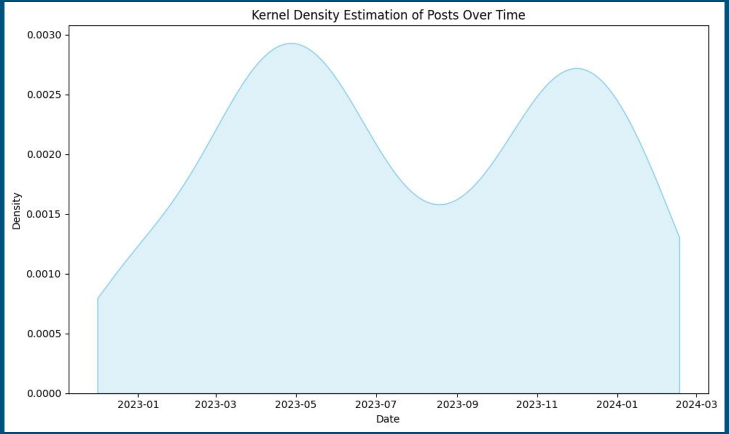
**We looked for hashtags but we did not obtain interesting results, so we focused our network on the words contained in the posts.**

# Temporal division

we divided our data-frame in two sub data-frames by the date of introduction of chat gpt: 30/11/22

#posts pre: 452,  #posts post: 248





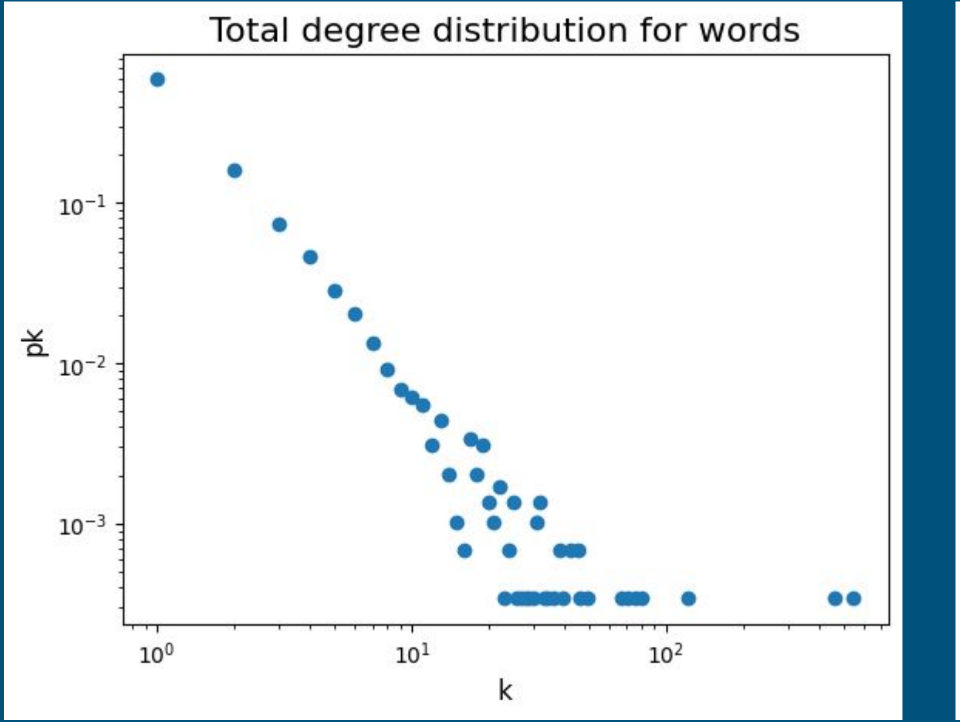
# Top cited dates

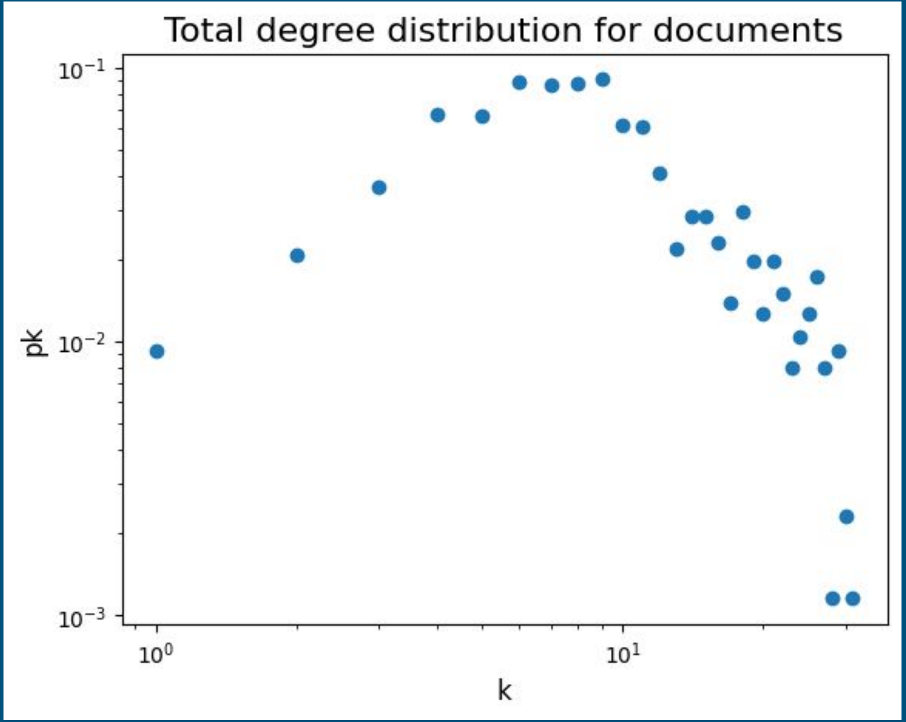
We looked for the dates in which the discussion about AI was intense for the post ChatGPT period:

1. 23/11/23 and 22/11/23 that correspond to the event: following the resignation of the Board of Directors of OpenAI, Sam Altman returns to lead the company;
2. 1/05/23 and 2/05/23 Geoffrey Hinton, one of the main founder of modern AI leaves Google;
3. 17/02/24 Chat GPT announce the new ability to memorize past chat with users.

Regarding the pre ChatGPT period the most important date is 27/01/2014 and corresponds to the Google announcement of the acquisition for 500 million dollars of Google DeepMind

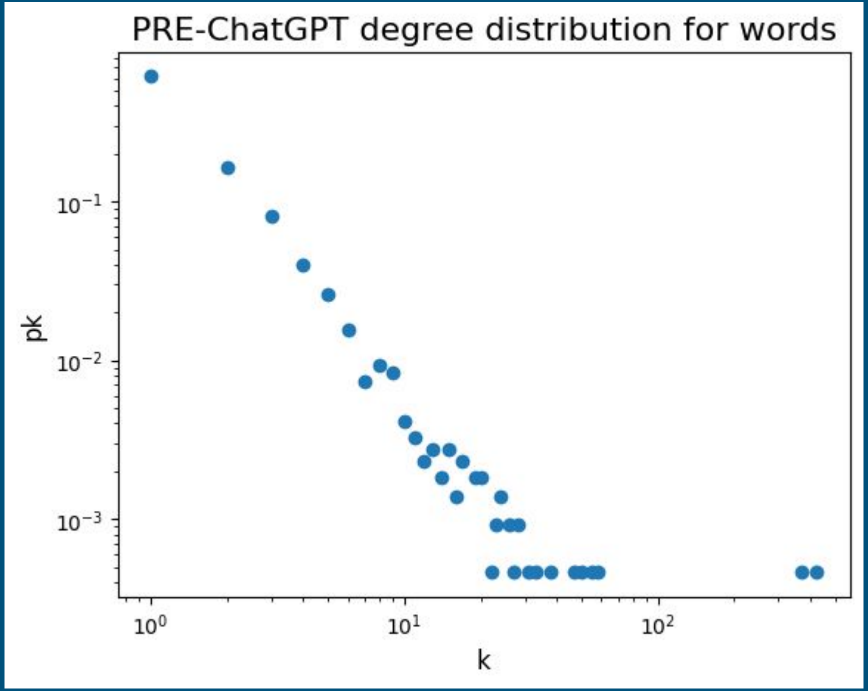
# Global Degree distribution

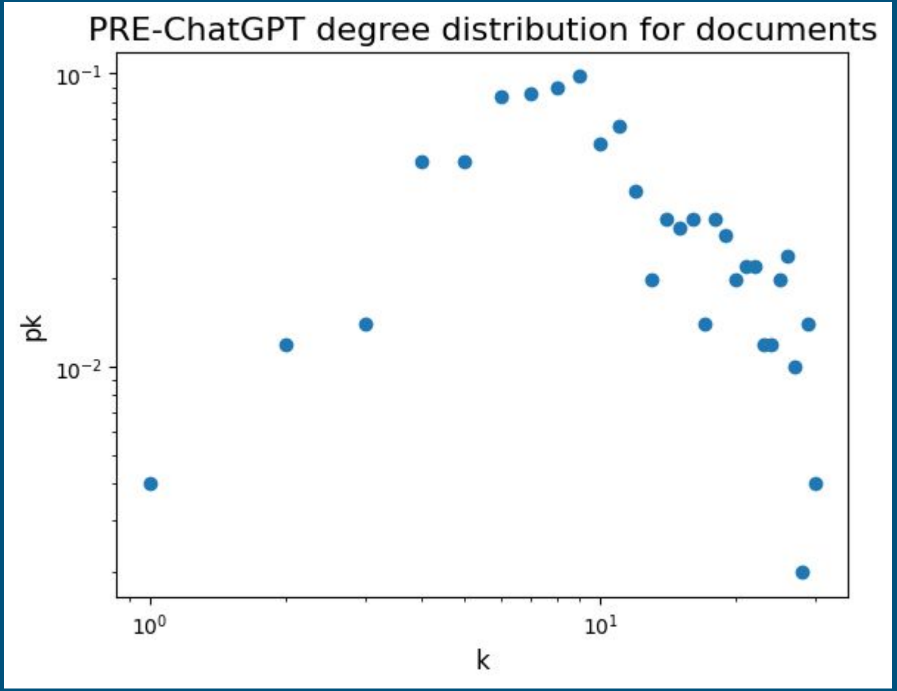




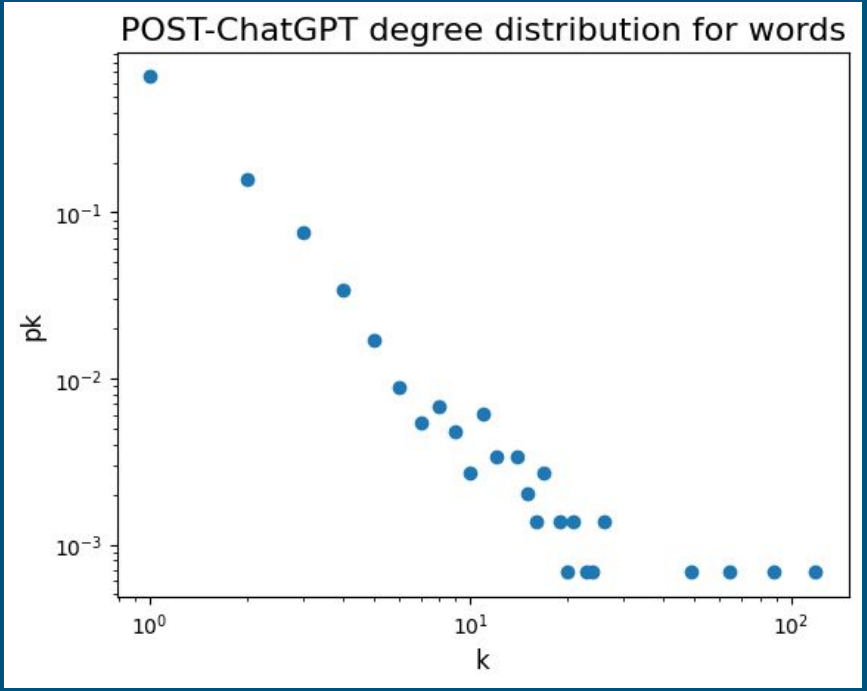
log log plot of the cumulative probability

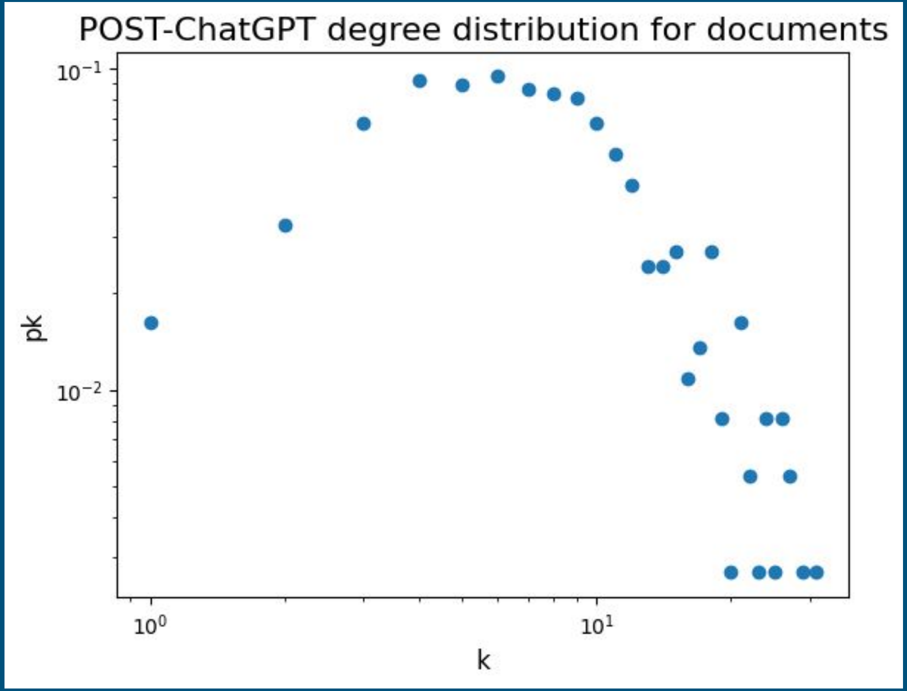
# Pre-Chat GPT Degree distribution





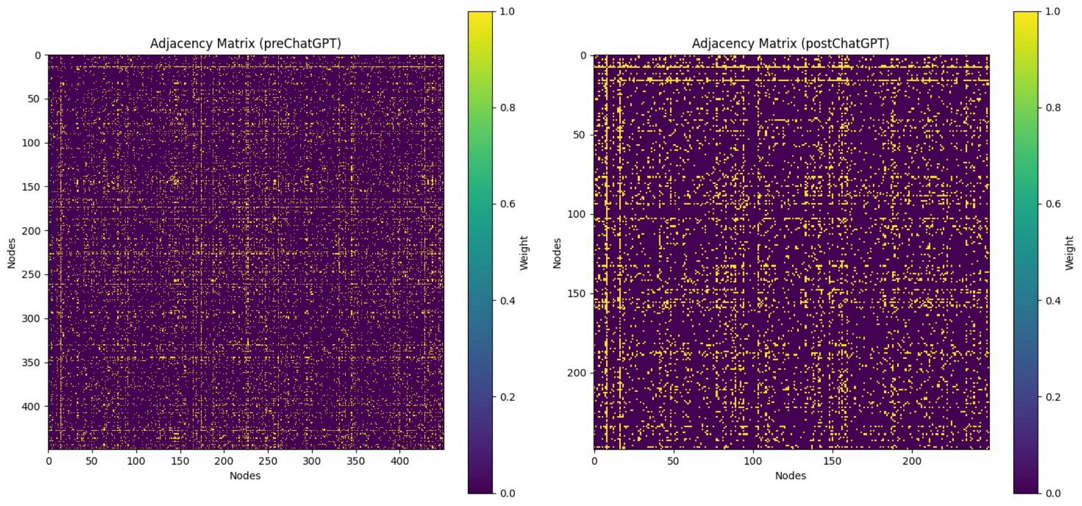
# Post-Chat GPT Degree distribution


POST-ChatGPT degree distribution for words

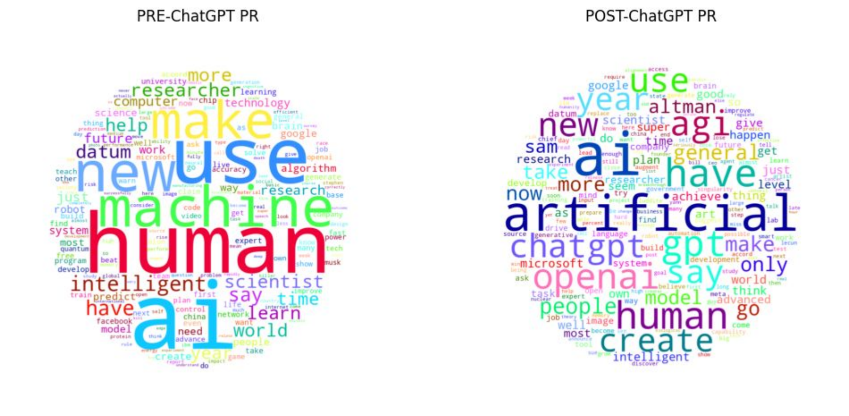

POST-ChatGPT degree distribution for documents

# Graph Creation and adjacency matrices

We created the graphs in two steps: first we added words and documents as nodes of a bipartite graph and then we extracted the projection of the bipartite on words or document for a final graph with words ad nodes connected if they belong to the same document.

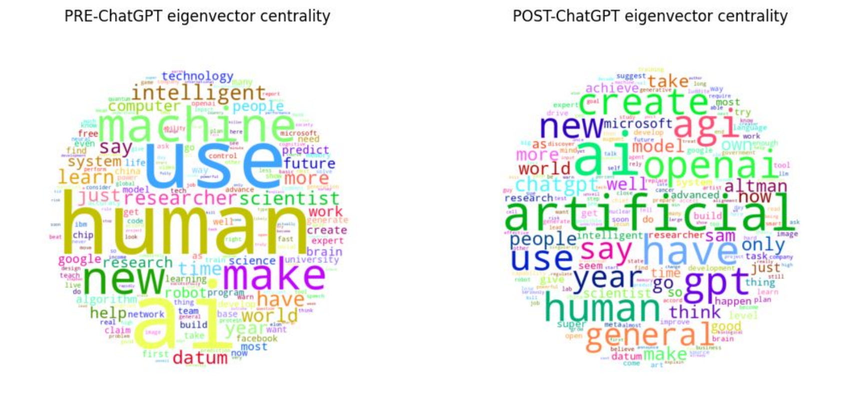


# WordCloud representation (1)



PRE-ChatGPT PR

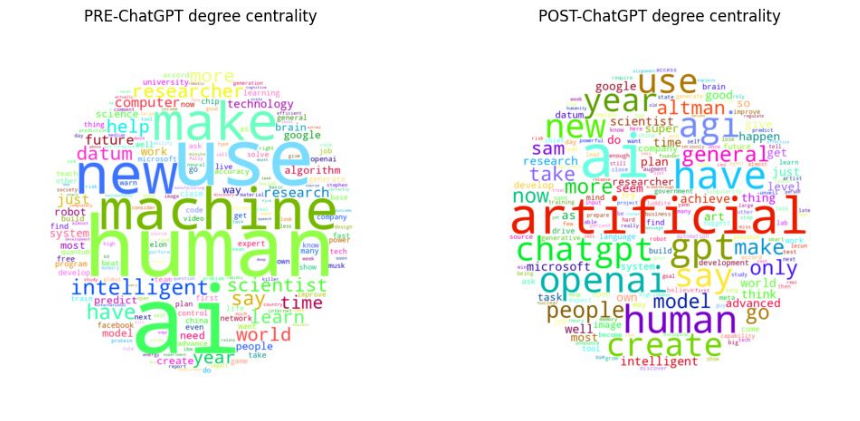
POST-ChatGPT PR



PRE-ChatGPT eigenvector centrality

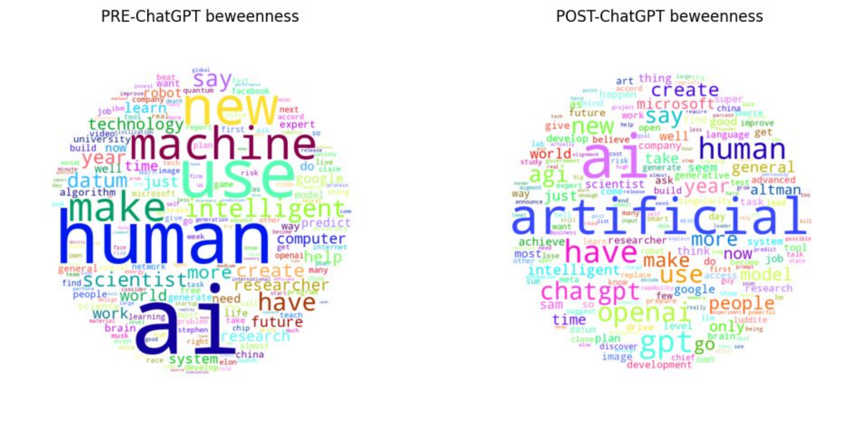
POST-ChatGPT eigenvector centrality

# WordCloud representation (2)



PRE-ChatGPT degree centrality

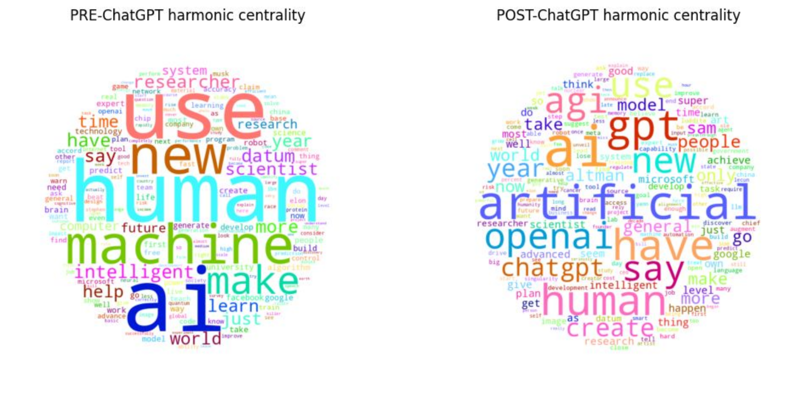
POST-ChatGPT degree centrality



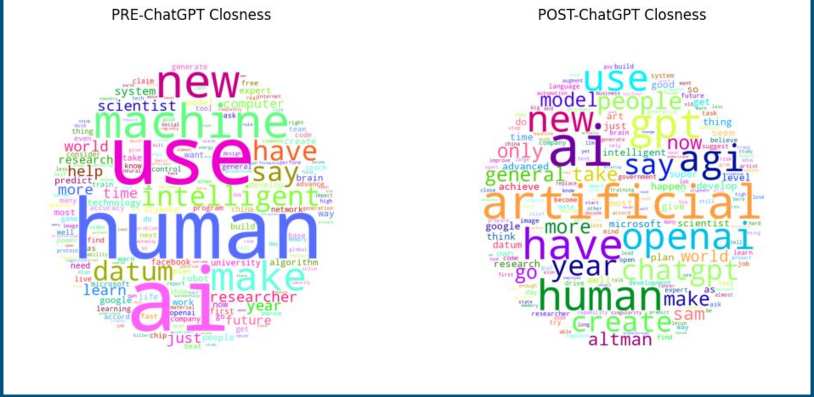
PRE-ChatGPT beweenness

POST-ChatGPT beweenness

# WordCloud representation (3)


PRE-ChatGPT harmonic centrality

POST-ChatGPT harmonic centrality


PRE-ChatGPT Closness

POST-ChatGPT Closness

# Graph Clustering Analysis

- **Global Clustering Coefficient**: The global clustering coefficient of the graph is 0.28 per the pre Chat GPT and 0.30 for the post chat gpt. This coefficient is a measure of the overall tendency of nodes in the graph to cluster together, forming tightly knit communities. A value closer to 1 indicates a high degree of clustering, while a value closer to 0 indicates a low degree of clustering. In this case, there is some clustering present in our network, but it's not very high.
- **Average Clustering Coefficient**: The average clustering coefficient is 0.48 for the pre and 0.52 for the post. This is the average of the local clustering coefficients for all nodes in the graph. It's quite a bit higher than the global clustering coefficient, which suggests that while the overall graph might not be highly clustered, there are many nodes that form tight communities.
- **Local Clustering Coefficients**: The local clustering coefficient for individual nodes varies significantly. Some nodes,  have a local clustering coefficient of 1.0. This means their neighbours form a perfect clique (every neighbour is connected to every other neighbour). On the other hand, other nodes have small  local clustering coefficient, meaning their neighbours do not connect at all.

While the overall network is not highly clustered, there are a number of nodes that exist within tight communities. This suggests the presence of subgroups within the network that are interconnected. The nodes with low clustering coefficients could be nodes that connect these communities or could be outliers in the network.

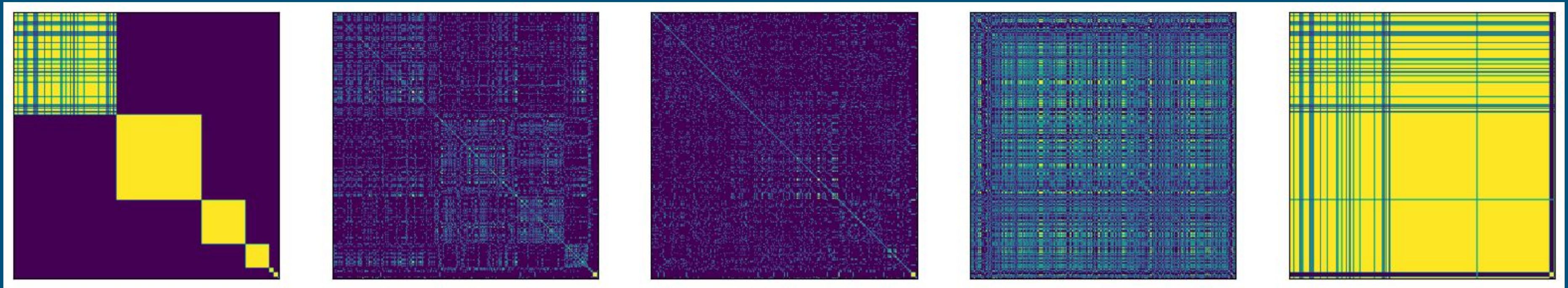
# Topic detection: the algorithms

1. greedy_modularity_communities of networkx library: employs an iterative greedy optimization approach to maximize modularity.
2. predefined Louvain method of the Python Louvain package
3. soft Louvain used during the labs
4. predefined Infomap of the Python Louvain package
5. BerTopic used during the labs

# Topic detection: the metrics

1. normalized mutual information NMI
2. modularity
3. ncut

# pre-Chat gpt community assignment matrices



greedy modularity communities

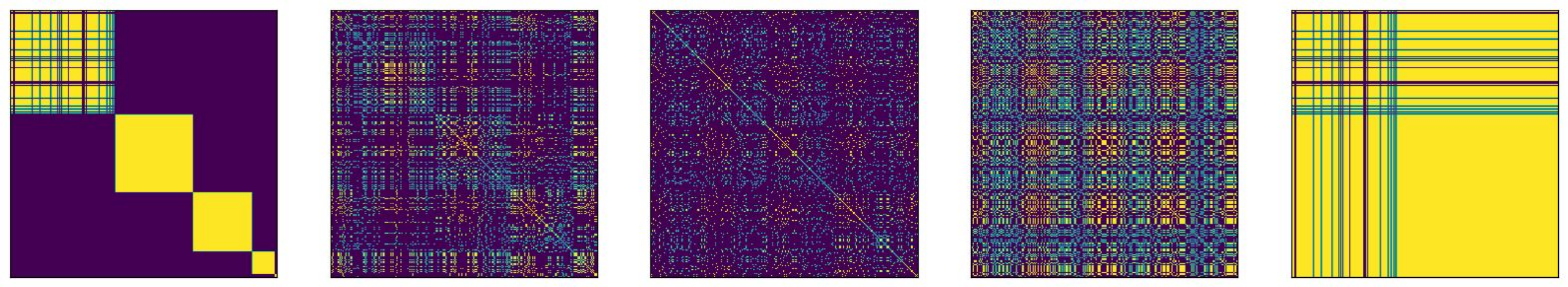
predefined Louvain

soft Louvain

BerTopics

Infomap

# post-Chat gpt community assignment matrices



greedy modularity communities

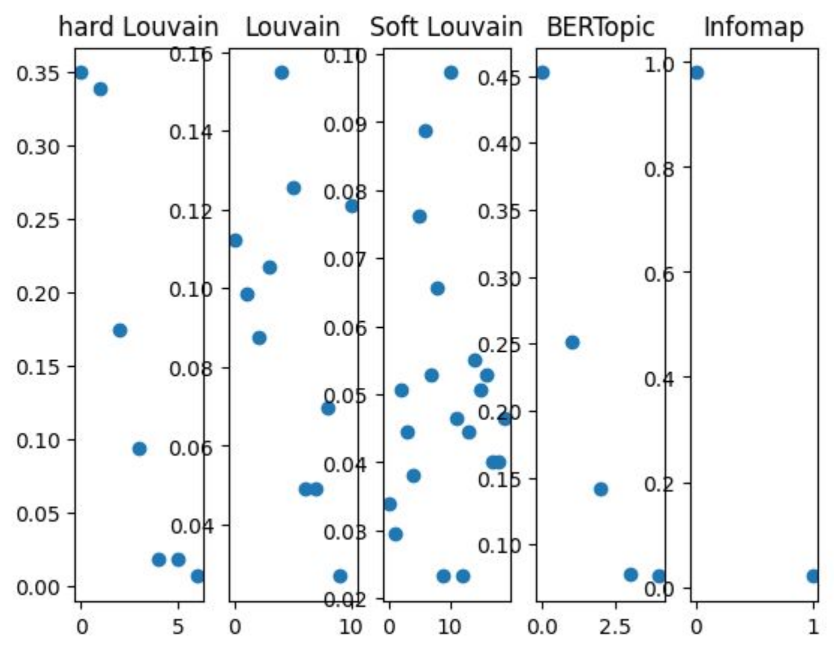
predefined Louvain

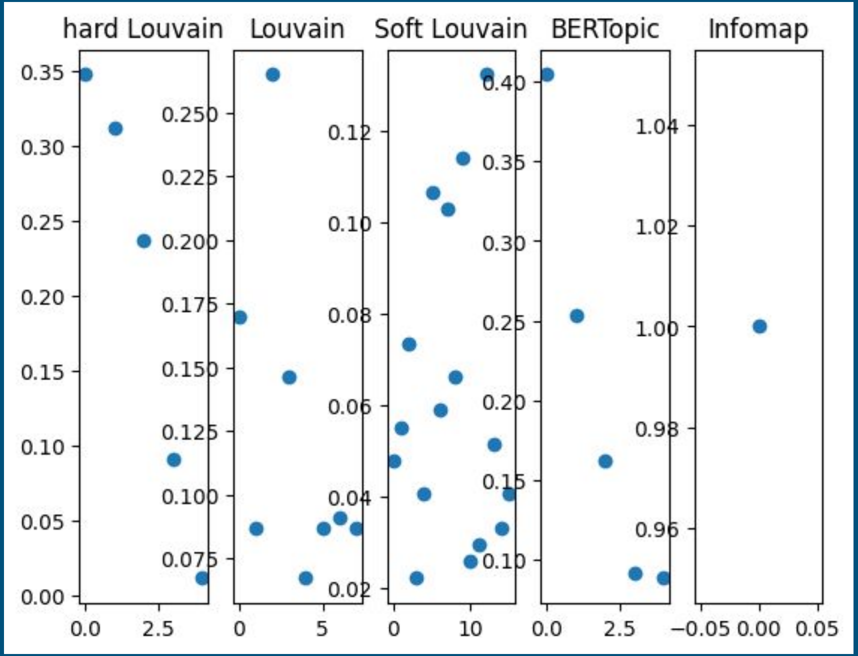
soft Louvain

BerTopics

Infomap

# % of words per community





# Results comparison in pre-Chat GPT

| Algorithm | # communities | NMI | Q | Ncut |
|---|---|---|---|---|
| Networkx Greedy Communities | 7 | 0.72 | 0.29 | 0.47 |
| predefined louvain | 11 | 1.36 | 0.33 | 0.55 |
| soft louvain | 20 | 1.6 | 0.47 | 0.47 |
| infomap | 2 | 1.17 | 0.11 | 0.21 |
| BerTopic | 5 | 0.64 | 0.15 | 0.62 |

# Results comparison in post-Chat GPT

| Algorithm | # communities | NMI | Q | Ncut |
|---|---|---|---|---|
| Networkx Greedy Communities | 5 | 0.81 | 0.27 | 0.47 |
| predefined louvain | 8 | 1.26 | 0.30 | 0.51 |
| soft louvain | 16 | 1.48 | 0.46 | 0.46 |
| infomap | 1 | 1.00 | 0.18 | 0.00 |
| BerTopic | 5 | 0.73 | 0.18 | 0.60 |

# Sentiment analysis-1

Two questions:

-“Which might be the average opinion of the users on this topic?”

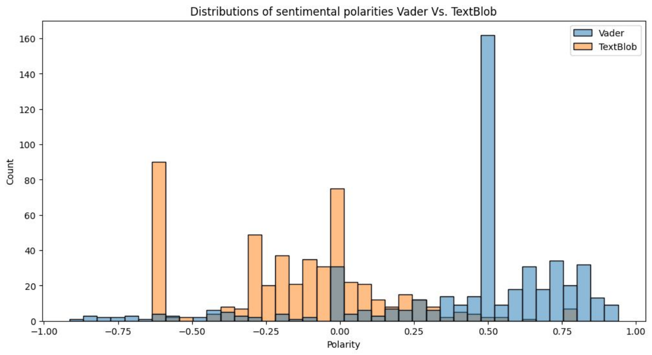
-“Can we infer some differences before and after ChatGPT?”

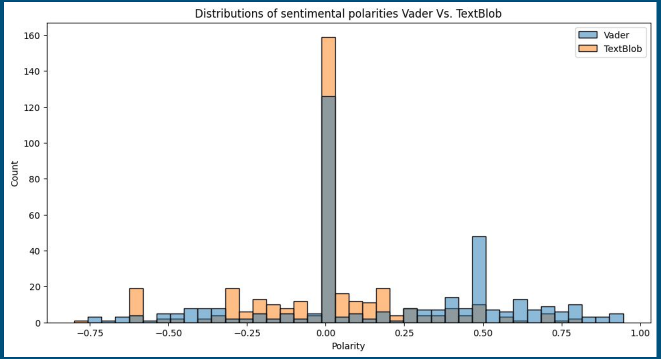
Two Methods:

-Vader analysis

-Blob Polarity

# Sentiment analysis-2



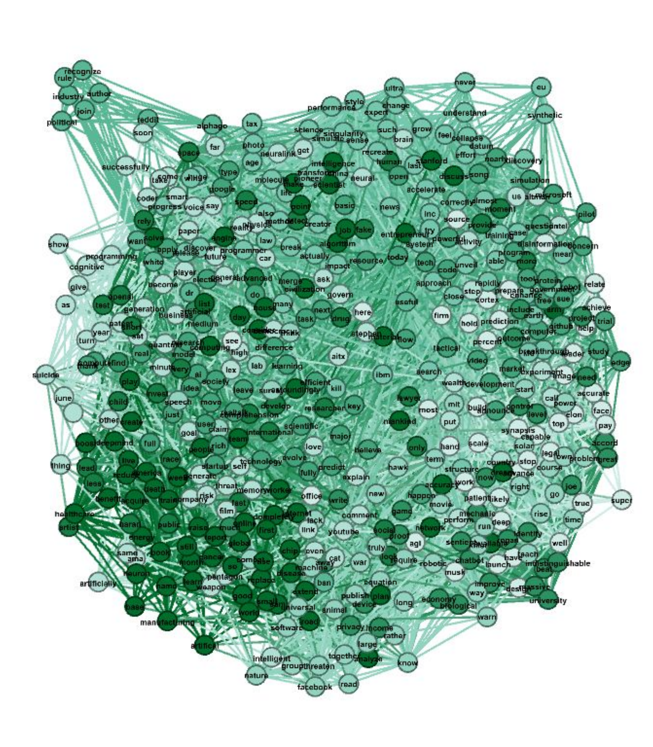


-*Before the event:* VADER analysis indicated a predominance of positive sentiments, while BLOB POLARITY highlighted a prevalence of negative sentiments.

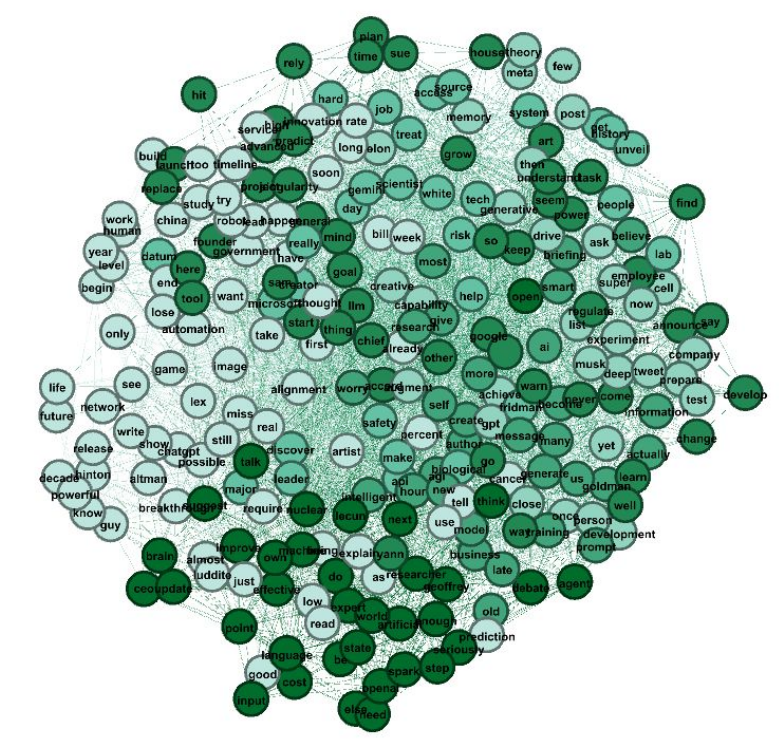
-*After the event:* both analyses showed shifts towards increased neutral and negative sentiments alongside decreased positive sentiments.

# Metrics comparison: modularity

pre ChatGPT

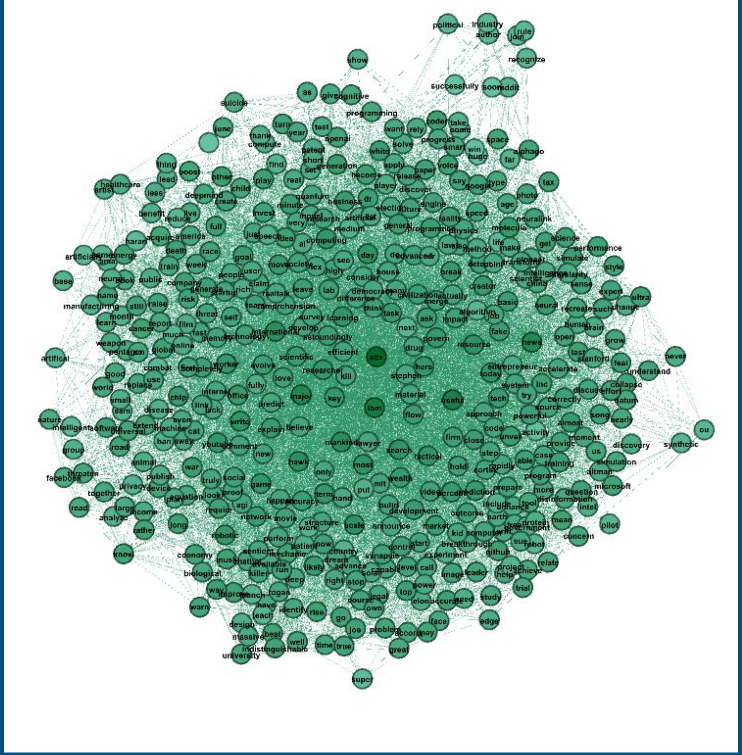


post ChatGPT

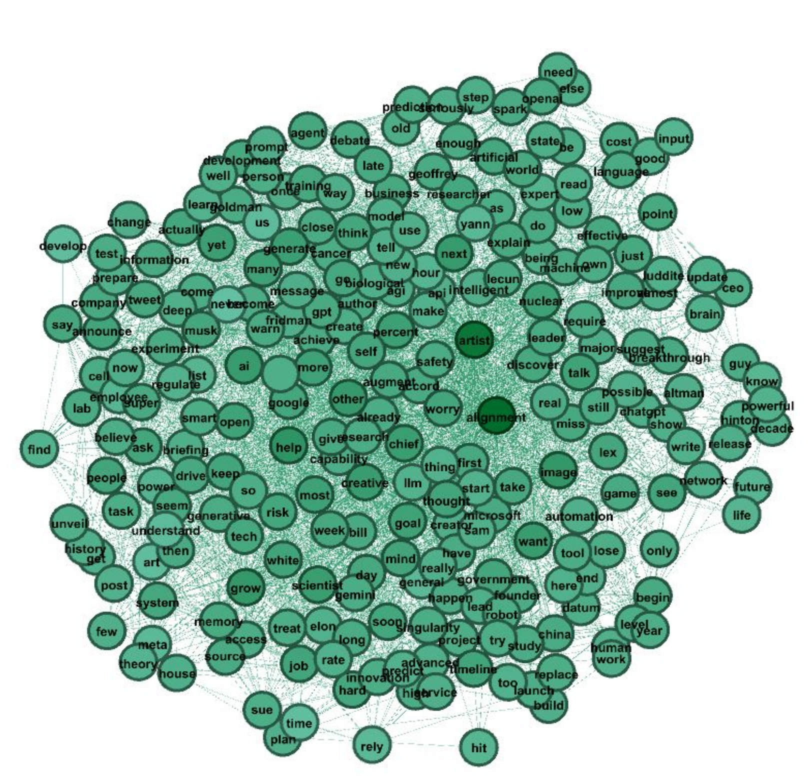


# Metrics comparison: closeness

pre ChatGPT

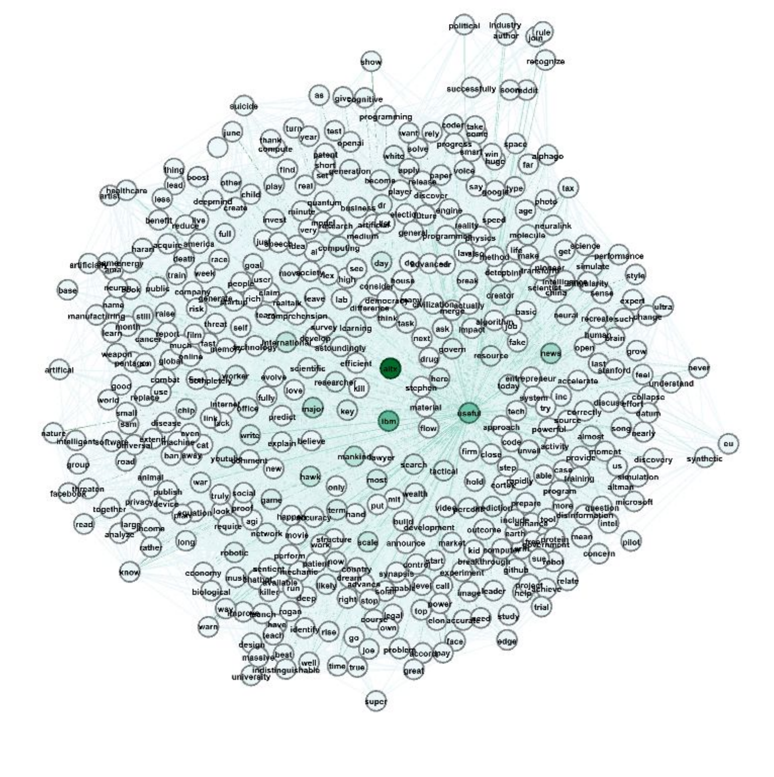


post ChatGPT

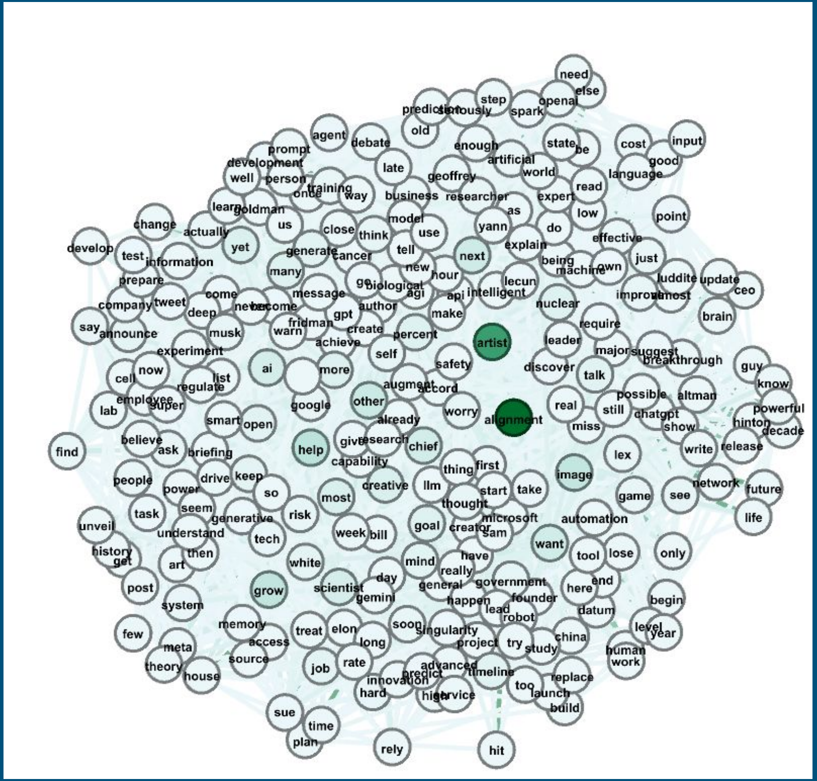


# Metrics comparison: betweenness

pre ChatGPT

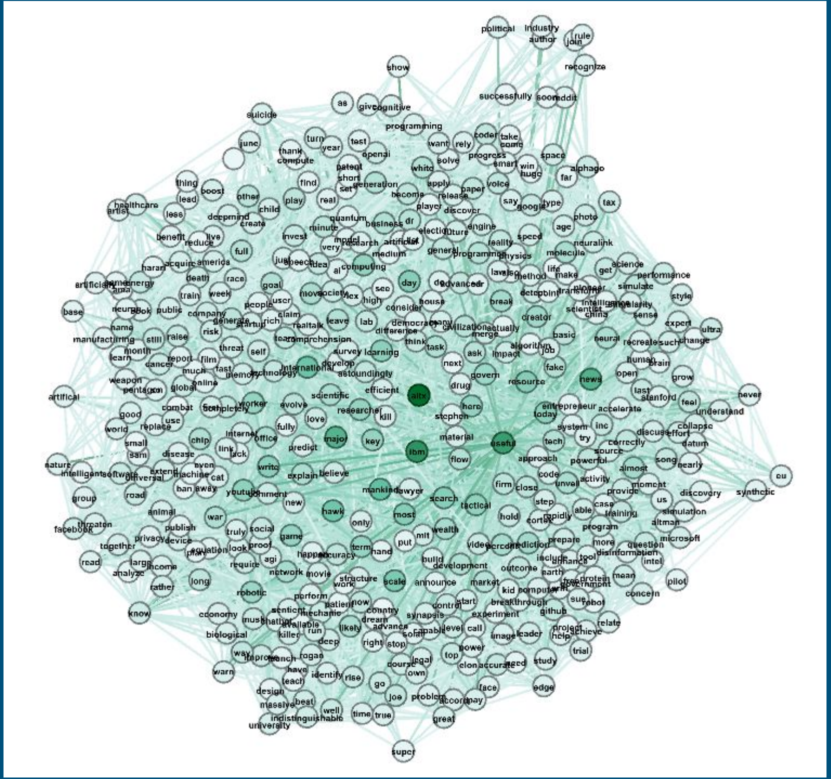


post ChatGPT

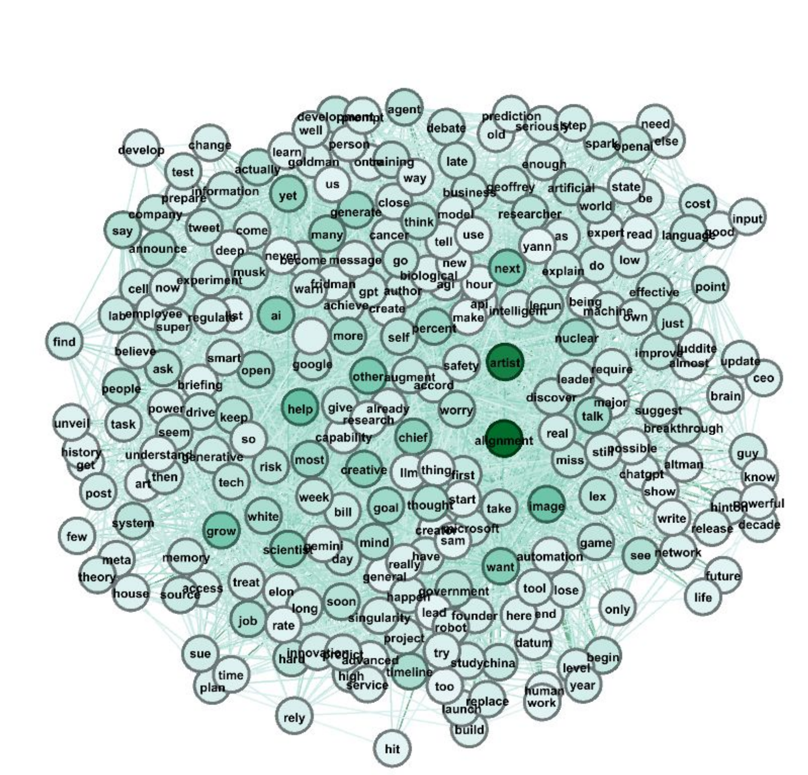


# Metrics comparison: PageRank

pre ChatGPT

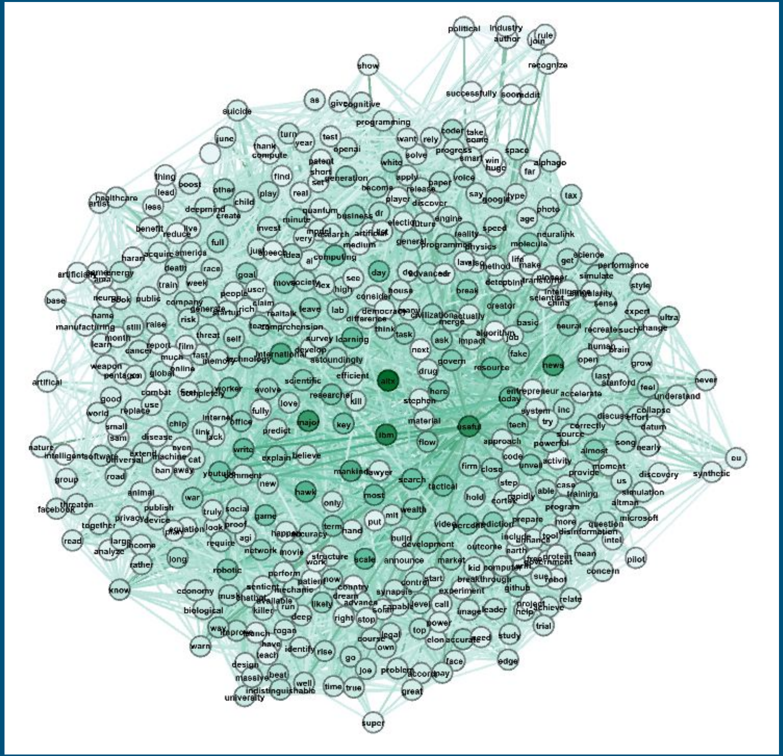


post ChatGPT

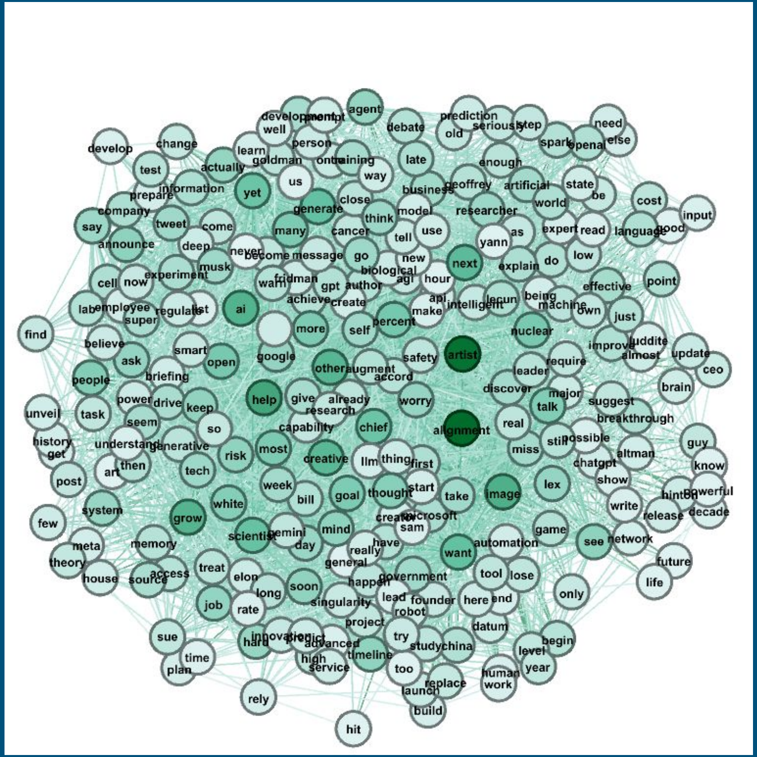


# Metrics comparison: HITS centrality

pre ChatGPT



post ChatGPT



# Thank you for your attention!
# Any questions??

Filippo Candeleri
Cecilia Rossi
Anna Zorzetto