Influence of Popular Trends in Social Communities

Anurag Jaiswal (2021CSM1002)
Arunava Chaudhari (2021CSM1020)
Shourya Gupta (2021CSM1007)

Abstract

Social media news is new form of news and it has impact greater than conventional news as people like to watch news online rather than watching news on TV.This saves time but has lot of effects in society.People are more updated and some news may affect lot of people and we need to study frequency of people or community of people where news is having impact and are these communities static or dynamic in all phases whether it be initial phase or intermediate phase or convergence phase.How communities evolve over time or do all communities of news survive or there are some news communities which dies off we need to observe this in our experiment.

1 Introduction

Online news media has grown to be a reliable and widely accessed source of information about the latest happenings worldwide. With the escalating popularity of smartphone-enabled internet browsing in recent years, a significant number of internet users currently acquire news from these sources. Over the years, these newspaper headlines tell us about some insights happening in social platforms, everyday life. This is interesting how such popular trends or news produce effect on social community in real world. Is it possible that popular trends divide people and how the interest of people change with popular trends? Talking about news headlines, they are mainly summarized, audited textual information and represent the fundamental idea of the corresponding news article. Such news headlines help study relationships between various news concepts and analyze the temporal dynamics of the news concepts and their relationships over time. Such popular trends will also able to change the person interest. This research is all about detecting trends in news headlines, or else we can say popular topics in the news in the media that people read. These topics will work as influential nodes in a network of people, and we will analyze how they create communities in a network and how they grow over time.

In this report we evaluate such popular trends or news from news media platform and spread it over the bunch of social people via high influencers and analysed its effect over the world to see how their interest of focus are changed.

1.1 Motivation

As any news platform acts critical in a country and people who are coming in contact with such news their focus of interest changes with respect to that news. We wanted to see how such popular news spreads their effects all over the country and is there any pattern to it and if such a pattern is biased or not. The entire motivation comes from the fact at the time of election several news are spread about the political parties across the country. How one of the popular news related to one of the parties changes the interest of the public towards that particular party and also other parties.

1.2 Literature Survey

Enhanced news sentiment analysis using deep learning Method by Wataru Souma · Irena Vodenska· Hideaki Aoyama. This paper establishes a connection between whether financial news is positive or negative and its impact on the stock price. Our goal is a little bit different. We have to analyze the impact of information in different communities. Deep learning methods in the paper may help us in establishing a connection of news in a community.

1.3 Problem

We wanted to analyze how popular news or trends over a period of time produces its effect on a social group of people. How the news influence people and form communities in the real world and changed their interest towards that.

We wanted to see how social media news influences people and change their interest over several periods of time. In general, consider a case where some news was spread by some different political parties, hence over the period of time how this news is going to change the interest of people towards these political parties.

1.4 New Experimental Idea

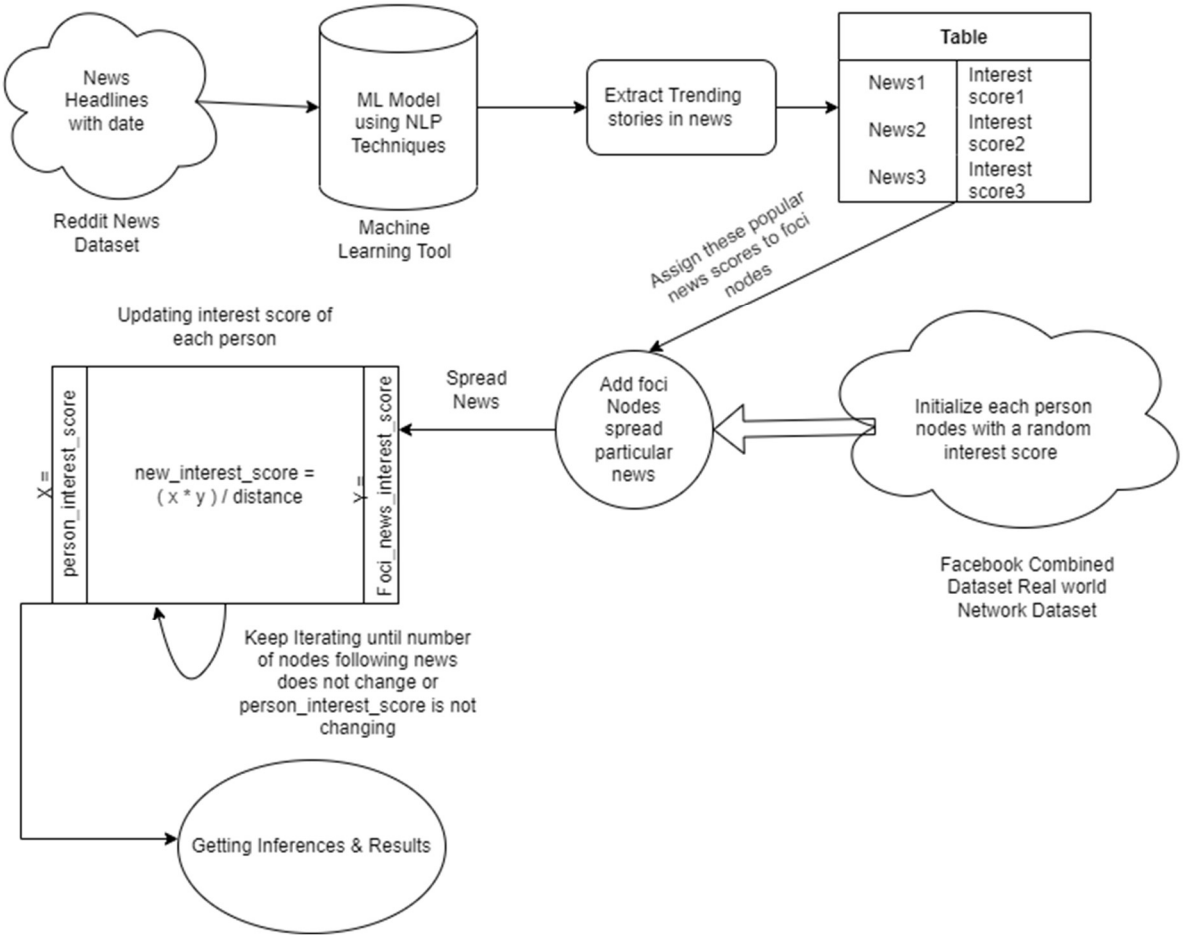


Fig: Flowchart Model

2 Method

We have divided our experiment into two parts. At first, we tried to extract the trending topics from the online news headlines. After that in the second part, we have chosen the top 10 trending topics from the above experiment and then use those distinct topics as foci nodes in the next experiment to see the influence of those topics in social communities.

2.1 Datasets Used

We have used Reddit News dataset for extracting the trending topic from online news headlines which are collected between a particular time duration. For visualizing the community structures, we have selected the Facebook network dataset. Description of the datasets and extracted information from the datasets are given below.

2.1.1 Reddit News Dataset

This dataset is used mainly for generating trending news from online news media which will act as a foci node in social community structure for further experiment. We have used Reddit News Dataset for extracting trending topics from the news headlines available on the

internet. This dataset has almost 74k news headlines from the year 2008 to 2016. Each row containsthe news headline and the on which it was published. Here at most top 25 news headlines from each day and the top news are chosen by reddit user's vote. The trending news extraction process is discussed below.

2.1.2 Facebook Network Dataset

This dataset containsareal-world network consists of a finite set of vertices(or nodes) and a set of edges that connect a pair of nodes. It has 4039 nodes and 88234 edges. The detailed information as follows.

Nodes	4039
Edges	883234
Average degree	43.6910
Average Clustering Coefficient	0.6055
Diameter (longest shortest path)	8
90-percentile effective diameter	4.7

2.2 Extract Trending Topics in News Headlines:

In recent years we have seen million of blogs, news reports and headlines are published on internet every day. And here social media platform comes into play as it becomes the main source of news online to fulfil the requirement of information consumption of internet users.

But this news often come with duplicated or junk contents, so we have to effectively pre-process, organize and analyse the contents of the of this huge news collection for providing better reading experience in short period to a user.

We have divided this process of extracting trending stories in news in several subtasks which start with News cleaning, Keywords Extraction, News clustering and the last is Trending News Visualization. In the next section we will explain about all of the above-mentioned subtasks in detail. Before we start let's talk about the news sources and crawling.

2.2.1 New Cleaning: Online news often contains many unwanted texts, words from other languages, or provider-specific patterns etc. Those text features, if not cleaned during the early stage of the news extraction process, it may create noises to downstream tasks. We have simply removed non-English characters from the headline texts. We have also seen that news provider may have some patterns in its articles. So, we tried to remove those pattern phrases as they might be recognised as keywords in headline text.

2.2.2 Keywords extraction: The objective of this subtask is to select several keywords with term frequency to reflect the key information of the news headlines. These keywords are often taken from the named entities and noun phrases in the headlines. We have used spacy which is an open-source python library, capable of most NLP applications. It supports

fast NER including most entities such as Persons, Organizations, GPE (countries, cities, states), etc. It also supports the extraction of noun phrases. This library helps us to extract entities and noun chunks efficiently.

2.2.3 Keywords Scoring: We have considered the importance of keywords in the news with different weights depending on the keyword location, keyword type, and frequency of appearing. A keyword appearing in the title is assigned with more weight than that in the content. An entity keyword weights more than a noun chank in the same location. We have created a simple formula to calculate the score of a keyword is as follows:

$$Score_k = W_{type} \text{ x } T_{k,in\ title} + T_{k,in\ content}$$

2.2.4 Keywords Filtering: The objective of keywords filtering is to remove the unwanted words, misclassified entities, and symbols, such as stop words, date-times, month names, prepositions, adjectives, determiners, conjunctions, punctuations, emails, special symbols #, @, ^, *, etc. We have used Spacy NLP model to classify the keywords extracted from the headlines. Along with the keywords which are related to news providers' names and writing patterns are also removed.

2.2.5 Keywords postprocessing: The goal of this subtask is to link the entities with their alternative names or abbreviations to improve keywords similarity calculation in the next stage. There are Python libraries such as BLINK to verify the extracted entities with Wikipedia data for general purpose applications. For this 73 k+ article dataset, there are about 500k keywords extracted in total. The most popular keywords are shown below.

	Keyword	Count		Keyword	Count		Keyword	Count
0	US	10903	34	Donald Trump	484	68	Italy	297
1	China	4435	35	Mexico	474	69	stock	294
2	United States	2037	36	Airbus	466	70	stake	293
3	Chinese	1678	37	Volkswagen	453	71	production	292
4	Trump	1619	38	American	447	72	Oil price	290
5	company	1323	39	Nissan	444	73	profit	289
6	German	1119	40	European Union	437	74	California	287
7	Boeing	1093	41	Washington	432	75	head	283
8	Fed	850	42	Google	429	76	Renault	282
9	Britain	774	43	Beijing	428	77	Saudi	279
10	investor	774	44	CEO	419	78	government	275
11	EU	761	45	Japanese	417	79	coronavirus pandemic	275
12	deal	745	46	tariff	412	80	employee	274
13	British	703	47	business	410	81	S&P	268
14	Japan	679	48	country	404	82	Ghosn	263
15	French	679	49	economy	399	83	effort	260
16	Federal Reserve	679	50	chief executive	397	84	coronavirus outbreak	257
17	share	670	51	France	394	85	Ford	256
18	MAX	659	52	India	386	86	Uber	252
19	Europe	657	53	Russia	376	87	board	252
20	UK	653	54	demand	342	88	Deutsche Bank	251
21	Tesla	630	55	concern	332	89	Swiss	250
22	Germany	629	56	Canadian	331	90	report	248
23	plan	627	57	White House	327	91	General Motors Co	243
24	coronavirus	620	58	Iran	326	92	Italian	243
25	European	613	59	bank	324	93	New York	243
26	Huawei	597	60	cost	319	94	US stock	240
27	talk	581	61	OPEC	319	95	Fiat Chrysler	239
28	Facebook	555	62	market	315	96	Hong Kong	238
29	Wall Street	553	63	Brazil	315	97	customer	238
30	Apple	540	64	GM	315	98	number	237
31	Canada	534	65	Amazoncom Inc	312	99	pressure	235
32	Amazon	532	66	decision	305	100	Mexican	234
33	sale	529	67	Brexit	304	101	airline	231

The top 100 keywords account for about 24% of all extracted keywords, while the top 500 are about 40%. For simplicity, this article builds an entity linking table by a quick check of those top keywords. As shown in the figure, the highlighted keywords in the same colour

are referred to as the same entity. Subsequently, a simple lookup table is created to link those top keywords to their alternative names or abbreviations. The keyword extraction in this approach runs quite fast. With a 2.5GHz CPU and 8GB RAM PC, it took about 2 hours and 35 minutes to complete all 73k+ news articles. On average, it needs less than 0.1s to process one article.

2.2.6 News Clustering: After the extraction of weighted key words from all the headlines, next task is to cluster the news into topics. News headline clustering is different from conventional text classification. News headline clustering is unsupervised learning. Furthermore, for clustering news topic from huge news dataset, there is no fixed number of clusters. We know that KMeans clustering is not suitable for this application, so we have used the following steps to cluster the news headlines:

2.2.7 Keywords vectorization: Here we have tried to convert the keywords of articles into numerical representation. The vectorizer in this article considers the scores of the extracted keywords as the term frequency. Depending on the number of news articles to be clustered, the CountVectorizer on HashingVectorizer from the Sklearn library is used. HashingVectorizer is more efficient for large datasets.

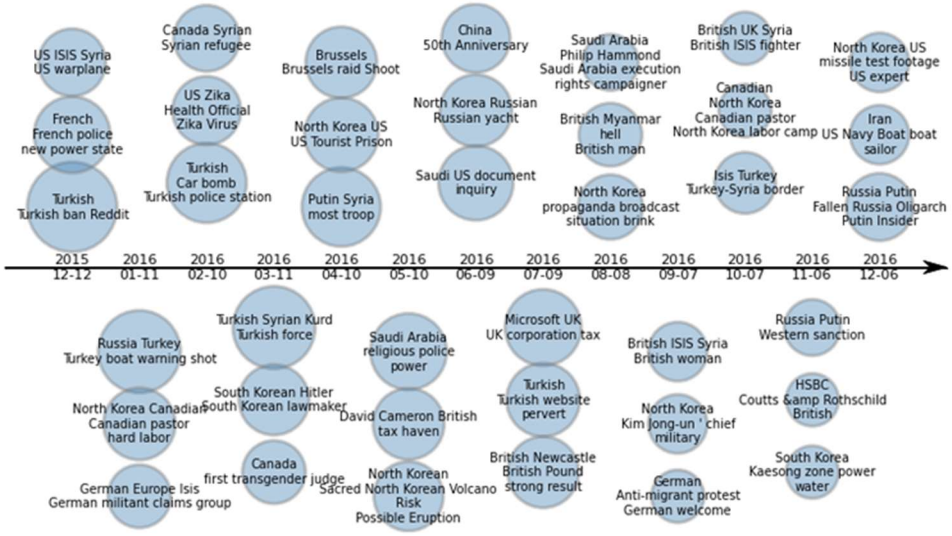
2.2.8 News Similarity based on Keywords: Next we have calculated the cosine similarity value between keywords extracted from two articles. The similarity matrix is then constructed for the list of articles to be clustered.

2.2.9 News Clustering: As we know that readers may have different views regarding the topics in news. So, there is now clear view on the number of topics. Here DBSCAN method is used to cluster the news headlines. Parameter eps is the maximum keyword distance (1-keyword similarity) of two news headlines in the same cluster. When eps approaches to 0, the clusters are more cohesive at the risk of clustering the same topic news into different clusters. When eps is 1, all news headlines fall into the same cluster. Here, eps is set 0.35 initially.

Along with that we have used another parameter max_size of the cluster as there are many headlines with top keywords (US, Chania, Tump etc), they may form super-sized clusters. The algorithm will further cluster the news in the super clusters using smaller eps until all cluster sizes are less than max_size.

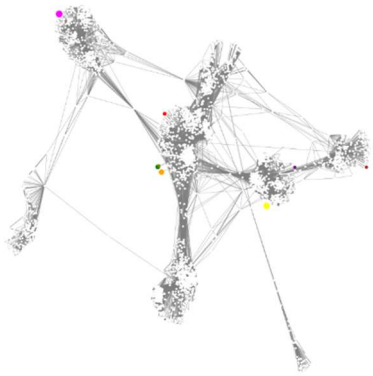
2.2.10 Trending stories visualization: Next news headlines are clustered; the trending topics are extracted straight forwardly. For any given date, the trending topics here are defined as the top clusters with more news headlines published in the past certain time period like the past two or three years.

The sample top trending topics between December 2015 and June, 2016 extracted from this dataset are visualized in the following figure.

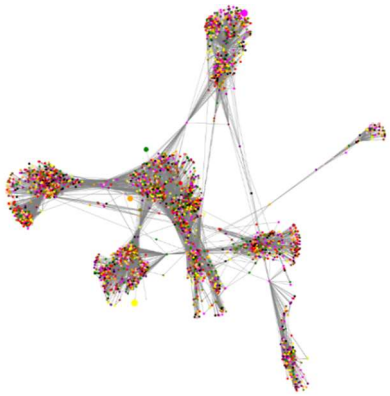


In this figure, the circle size is calculated by taking the logarithms of the number of articles in the cluster. The keywords in the circles are the keywords for those trending topics in a particular time period.

2.3 Analysis of Facebook Combined Dataset



Add foci nodes to graph



Initial Randomized interest of person node

We analyze the effect of such popular news on the real-world social network dataset such as Facebook combined dataset. We added 10 foci nodes as news spreader nodes. Normal node assumed as a person node and initialized with random interest score for each foci node. We apply those popular trends or news with normalized scores generated by the model to the foci nodes.

For evaluating the new_interest_score of each node with respect to each foci nodes we used three parameters.

1. X = foci_node_interest_score(comes from machine learning model)
2. Y = person_node_interest_score(intialise random interest score of each node for all foci nodes)
3. Distance = shortest distance path length from foci node

4. New Interest Score = (X * Y) / Distance (Calculating interest score for each node with respect to each foci node)

As new interest score is evaluated for a node with respect to foci node. So it will increase its interest_score by that amount and simultaneously decrease the (new_interest/k) interest value for other foci nodes.

Keep Iterating for each nodes until their interest scores are not changing

```
count_interest_change = []
rounds=1
while(True):
        before=no_of_nodes_following_news(G)
        G = spread_news(G)
        G,interest_change_list = update_interest_score(G)
        after=no_of_nodes_following_news(G)
        # print('frequency of person following news',after)
        # print('interest_change_list',interest_change_list)
        count_interest_change.append(len(interest_change_list))
        if(compare_dicts(before,after)==True):
        break
        print('Finished round',rounds)
        rounds=rounds+1

    # if rounds % 20 == 2:
    #     G = update_colors(G)
    #     plot_graph(G)
        # swap_interest_plot_graph(G,interest_change_list)
```

Spread News

```
foci_nodes = get_foci_nodes(G)
# print('foci_nodes',foci_nodes)
def spread_news(G):
    for each in G.nodes():
        if G.nodes[each]['type'] == 'person':
            person_temp_scores = {}
            for fnode in foci_nodes:
                G = calculate_interest(G,each,fnode)
        # break

    return G
```

Spread news function basically calculate interest score for each node with respect to each foci node.

Updating the interest score after each iteration.

```
def update_interest_score(G):
    # final_updates = {}
    interest_change_node = []
    for each in G.nodes():
        if G.nodes[each]['type'] == 'person':
            interest_score = G.nodes[each]['interest_score']
            tmp_score = G.nodes[each]['tmp_score']
            prev_fnode = max(interest_score, key= lambda x: interest_score[x])
            prev_color = G.nodes[prev_fnode]['color']
            for key in interest_score:
                if key in tmp_score:
                    interest_score[key] = interest_score[key] + tmp_score[key]
                else:
                    pass
            # print('interest_score',interest_score)
            curr_fnode = max(interest_score, key= lambda x: interest_score[x])
            curr_color = G.nodes[curr_fnode]['color']
            G.nodes[each]['interest_score'] = interest_score
            if prev_color != curr_color:
                interest_change_node.append((each,prev_fnode,curr_fnode))
    return G,interest_change_node
```

2.4 Technologies Used

2.4.1 Programming Language

We used python programming language for the experiments.

2.4.2 Libraries Used

- Numpy,Pandas
- NLTK, Spacy for text filtering
- JsonLoader,Seaborn,Matplotlib,Calendar
- Unicode Data,funct tools
- Networkx

3 Results

3.1 Experiment Findings

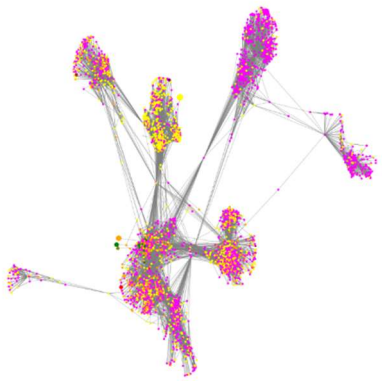
Network converges after some time.Time unit can day month or year as per the needs

Interest of people isfluctuating about news this is due to presence of friends and how influential is news

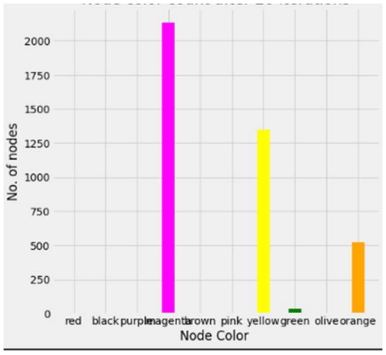
Communities generated through news are not stationary in the initial and intermediate phases and this change in communities cannot be predicted in these phases as this process is purely random.This process resembles how news is accepted by people and as the time progresses people do change decision but after a long time

3.1.2 Facebook Combined Real world dataset Inferences

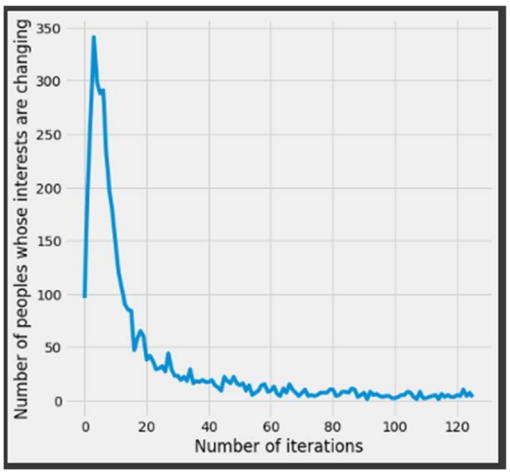
Graph is settled after some time and some news vanishes over the period of time. In the intermediate stage people are attracted towards those news which vanishes in the end.Some news communities are very small (they don't vanish) and some communities following a news is very large. As for facebook combined dataset we got 2-3 major news.



Interest score of each node after convergence

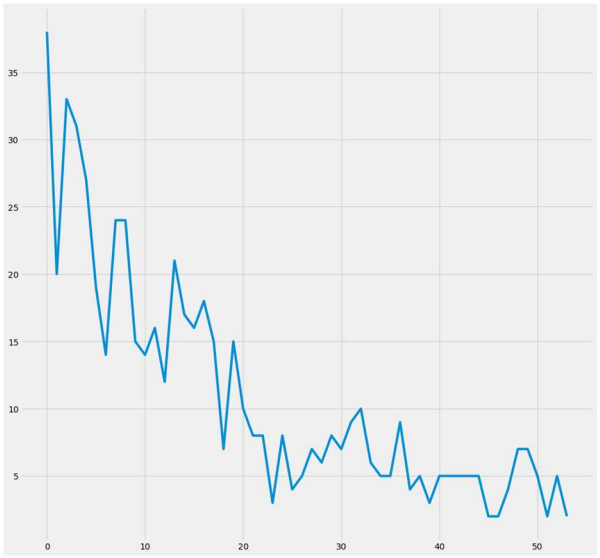


plot for no. of nodes vs node color



In second execution of same graph no of iterations in which graph converges is 100 different from 50 iterations also intermediate behavior is not same which validates our fact influence of news on a network until it settles cannot be predicted so pattern of news spreading cannot be predicted but after convergence communities can be easily detected which is difficult in intermediate stage as no of people in communities of news varies irregularly.

People are indecisive when news spread some fluctations in no of people happens when news spreads and rate of fluctation is different not same for all cases.

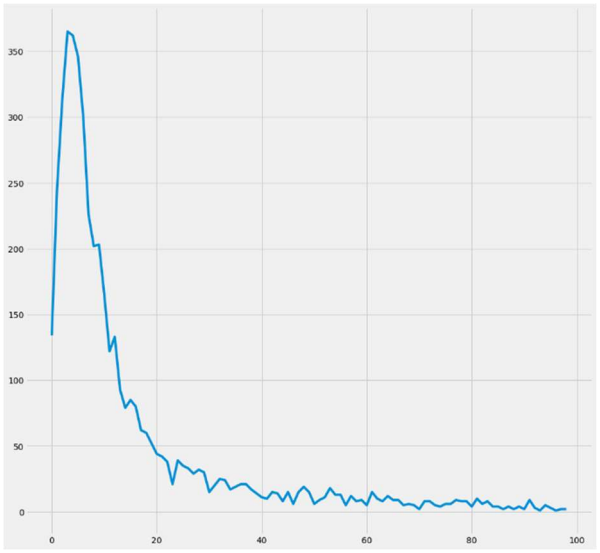


Facebook graph converges in nearabout 50 iterations in first time

X axis denotes the number of iterations and Y axis denotes no of people whose are changing

When this experiment is conducted second time resultant plot is different

Plot of second execution is given below

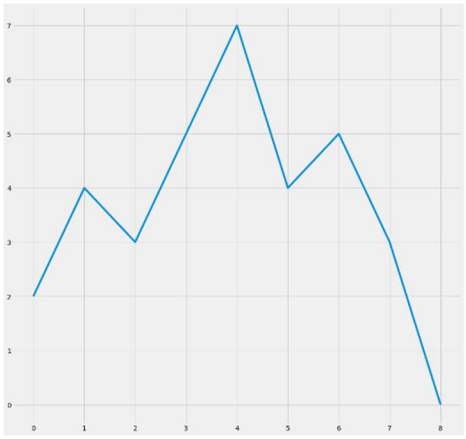


X axis denotes the number of iterations and Y axis denotes no of people whose are changing

In second execution of same graph no of iterations in which graph converges is 100 different from 50 iterations also intermediate behavior is not same which validates our fact influence of news on a network until it settles cannot be predicted so pattern of news spreading cannot be predicted but after convergence communities can be easily detected which is difficult in intermediate stage as no of people in communities of news varies irregularly

3.1.3 Observation of News Spreading Behaviour on Random graphs(Generated through Networkx)

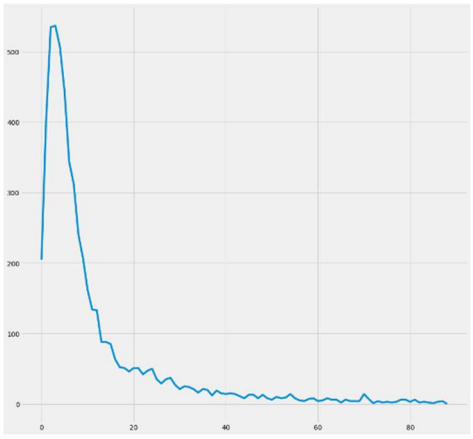
Plot of graph with 50 nodes and less edges



X axis denotes the number of iterations and Y axis denotes no of people whose are changing

This plot also converges but intermediate behavior is also irregular here.No of iterations for convergence also changes if we execute the same experiment second time.

Plot of graph with 5000 nodes and edges above 24 lakh



X axis denotes the number of iterations and Y axis denotes no of people whose are changing

Here also graph is converging in nearly 90 iterations so we can establish the fact the convergence of graph is not dependent on no of edges because facebook graph has 1 lakh edge and in second time it converges in nearly 100 iterations.

Convergence does not depend on no of nodes because facebook graph has nearly 4100 nodes(executed 2nd time) but is converging after the graph with 5000 nodes .On the other hand graph with 50 nodes is converging before facebook graph

4. Conclusion :

News communities graph has converged. Some news are accepted by large no of peoples and some news are accepted by very small no of people and some news is not accepted by anyone .Our simulation is nearly resembling real world scenario where some news in the long run is accepted by lot of people and some are forgotten by all and some news stayes in memory of some people

5.Future Scope and Improvements

We can use this model to further study how fakenews spread by linking fake news to some of the near neighbours of the influential people .Influential people are people with higher friends .We can pump fake news by increasing its score abruptly and then observe the effect

Random score can be replaced by real time interests and their dependencies with other interest can be analysed and random score can be generated really by considering interests location and other factors