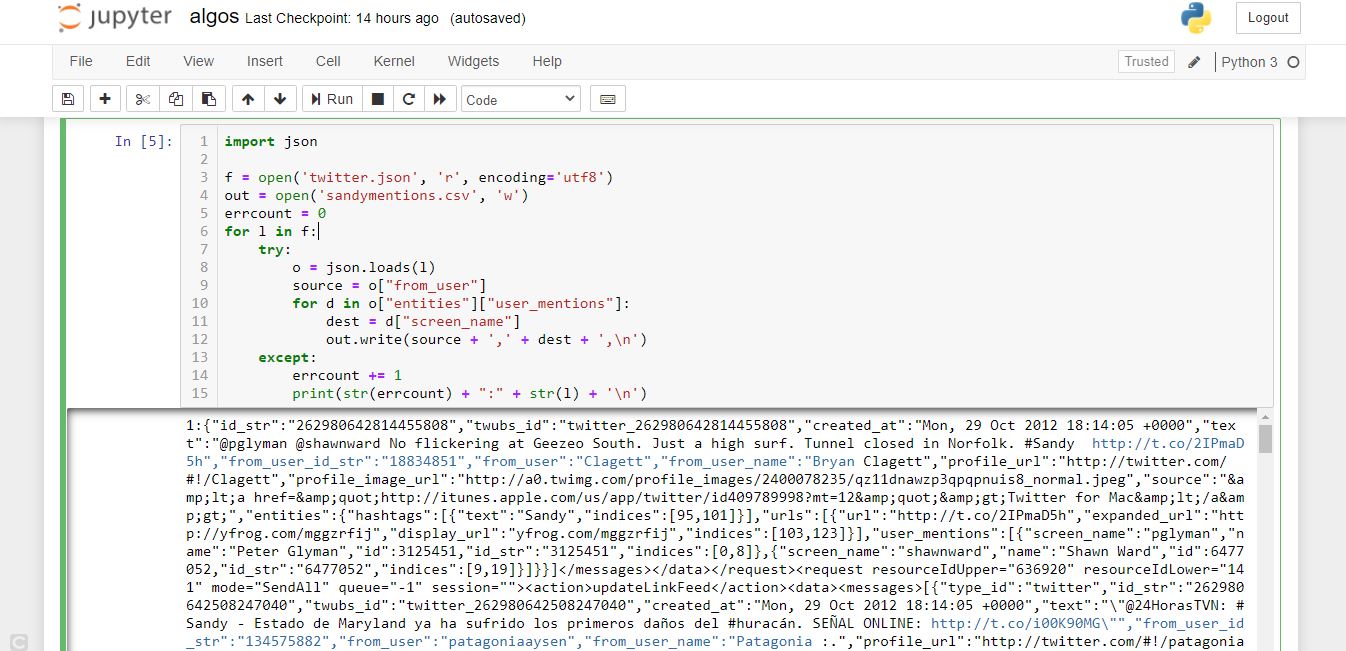
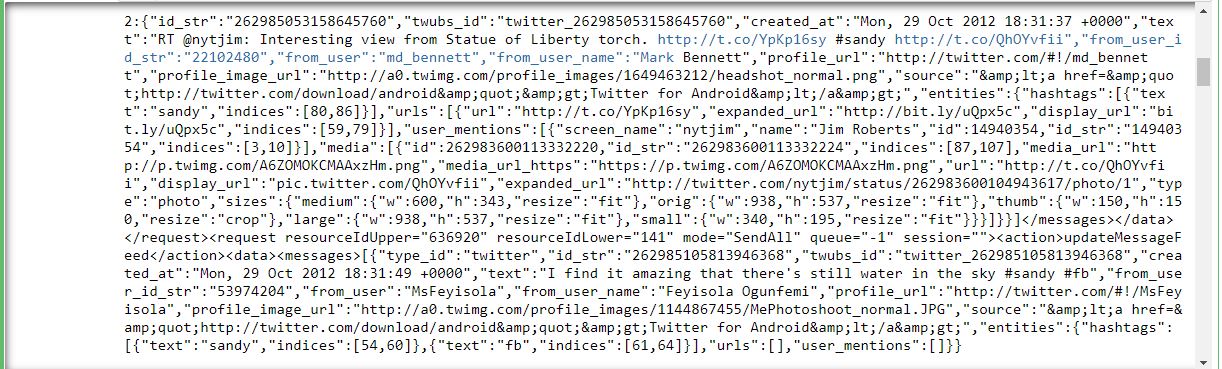
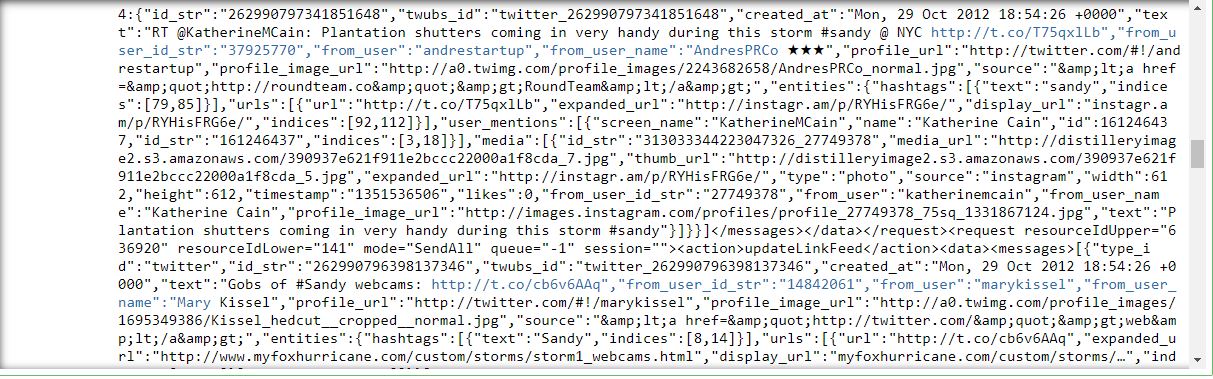
|  |
| --- |
| Project report  on |
|  |
| **Active Learning of Network Topology Properties** |
|  |
| **Project Guidance By**  **Dr. Qiong Cheng**  **Team details**   |  |  | | --- | --- | | **Harikrishnan Sundaram 801199594** |  | |
| **Dipriya Pillai- 801218036**  **Amulya Chepyala- 801204218** |
|  |
|  |
|  |
|  |
|  |
|  |
| **Introduction**  **Graph**  A Graph is a non-linear data structure consisting of nodes and edges. The nodes are sometimes also referred to as vertices and the edges are lines or arcs that connect any two nodes in the graph.  Ex:  In the above graph,  V = {0, 1, 2, 3, 4}  E = {01, 12, 13, 23, 34}  **Basic Operations**   * *adjacent(G, x, y)*: Checks whether there is an edge from the vertex x to the vertex y. * *neighbors(G, x)*: Lists all vertices y such that there is an edge from the vertex x to the vertex y. * *add\_vertex(G, x)*: Adds the vertex x, if it is not there. * *remove\_vertex(G, x)*: removes the vertex x, if it is there. * *add\_edge(G, x, y)*: adds the edge from the vertex x to the vertex y, if it is not there. * *remove\_edge(G, x, y)*: removes the edge from the vertex x to the vertex y, if it is there. * *get\_vertex\_value(G, x)*: returns the value associated with the vertex x. * *set\_vertex\_value(G, x, v)*: sets the value associated with the vertex x to v.   **Applications of Graph Data Structure**   * GPS Navigation * Transportation network * Utility Graphs * Protein-protein interactions graphs * Protein-protein interactions graphs * Networks routing * Electronic circuits * Telecom * Social networks. E.g., Facebook uses a graph for suggesting friends * Recommendations: Most Online portals uses it for recommendations   **The common terms in network topology** Bridges: A [network bridge](https://en.wikipedia.org/wiki/Network_bridge) connects and filters traffic between two [network segments](https://en.wikipedia.org/wiki/Network_segment) at the [data link layer](https://en.wikipedia.org/wiki/Data_link_layer) (layer 2) of the [OSI model](https://en.wikipedia.org/wiki/OSI_model) to form a single network.Switches: A [network switch](https://en.wikipedia.org/wiki/Network_switch) is a device that forwards and filters [OSI layer 2](https://en.wikipedia.org/wiki/OSI_layer_2) [datagrams](https://en.wikipedia.org/wiki/Datagram) ([frames](https://en.wikipedia.org/wiki/Frame_(networking))) between [ports](https://en.wikipedia.org/wiki/Computer_port_(hardware)) based on the destination MAC address in each frame.Modems: [Modems](https://en.wikipedia.org/wiki/Modem) (Modulator-Demodulator) are used to connect network nodes via wire not originally designed for digital network traffic, or for wireless.Hop Count: The number on nodes in between to specific nodes in the Hop count of the network. The shortest hop count of the network is the minimum number of nodes between two specific nodes.Firewalls: A [firewall](https://en.wikipedia.org/wiki/Firewall_(computing)) is a network device for controlling network security and access rules.Closeness: It is the average hop count between two specific nodes.Eccentricity: It is the maximum distance between a node to any other node in the network.Routers: A [router](https://en.wikipedia.org/wiki/Router_(computing)) is an [internetworking](https://en.wikipedia.org/wiki/Internetworking) device that forwards [packets](https://en.wikipedia.org/wiki/Packet_(information_technology)) between networks by processing the routing information included in the packet or datagram (Internet protocol information from layer.  1. **Diameter:** It the maximum eccentricity of a node over all the nodes in the network. 2. **Radius:** It the minimum eccentricity of a node over all the nodes in the network. 3. **Persistence:** It is the minimum number of links that to be removed from the graph to disconnect it. 4. **Girth:** It is the shortest hop count of closed loop in the graph. 5. **Expansion:** It is the average fraction of nodes in the graph that fall within a ball of radius centered at a random node in the topology. 6. **Betweenness:** It is defined as the number of shortest paths between pairs of nodes that traverse a node or link. 7. **Central Point of Dominance:** It the largest value of betweenness centrality if the network. 8. **Network Interfaces:** A network interface controller (NIC) is computer hardware that provides a computer with the ability to access the transmission media, and has the ability to process low-level network information 9. **Repeaters**: A repeater is an electronic device that receives a network signal, cleans it of unnecessary noise and regenerates it. The signal may be reformed or retransmitted at a higher power level, to the other side of an obstruction possibly using a different transmission medium, so that the signal can cover longer distances without degradation. 10. **Hub**: A repeater with multiple ports is known as hub, an Ethernet hub in Ethernet networks, a USB hub in USB networks. USB networks use hubs to form tiered-star topologies. Ethernet hubs and repeaters in LANs have been mostly obsoleted by modern switches.   **Network Topology Properties**  **Degree of a node and its distribution**  The degree is the number of edges that connect to a node. It is a fundamental parameter that influences other characteristics, such as the centrality of a node. The degree distribution of all nodes in the network helps define whether a network is scale-free or not, as we will see later. In the figure, the degree of each node is indicated and reflected in its size and color. Directed network nodes have two values for degree: out-degree for those edges coming out of the node and in-degree for those edges coming into the node.  A picture containing diagram  Description automatically generated  **Strength of a Node and its distribution**  If wij is the number of possible interactions between any ith and jth nodes, then the strength (si) of a node i is given by  The spread in the strength of a node has been characterized by a distribution function P(s); where    **Average Weight of a node**  If wij is the number of possible interactions between any ith and jth atoms, then the average weight < wij > of an atom i is given by  Image result for Average Weight of a node  **Characteristic path length of a network**  The characteristic path length (L) of a network is the shortest path length between two nodes averaged over all pairs of nodes and is given by  Related image   |  | | --- | | **Clustering** **coefficient of a network** | | The clustering coefficient (C) is a measure of local cohesiveness. Traditionally the clustering coefficient Ci of a node i is the ratio between the total number (ei) of the edges actually connecting its nearest neighbors to the ith node and the total number of all possible edges between all these nearest neighbors [ ki(ki−1) / 2 ; if the ith vertex has ki neighbors] and is given by  Image result for Clustering coefficient of a network The  **Random network**  In random networks any node has the same probability to be connected with any other node of the network. On the other hand, in scale-free networks, some nodes act as “highly connected hubs” (high degree), although most nodes are of low degree.  Network topology. A. The random network is homogeneous: most nodes (visualized as green dots) have approximately the same number of links. B. The scale-free network is heterogeneous: the majority of the nodes have one or two links (also commonly referred as edges) but a few nodes have a large number of links i.e. hubs (visualized as red dots), guaranteeing that the system is fully connected. Network topology. A. The random network is homogeneous: most nodes (visualized as green dots) have approximately the same number of links. B. The scale-free network is heterogeneous: the majority of the nodes have one or two links (also commonly referred as edges) but a few nodes have a large number of links i.e. hubs (visualized as red dots), guaranteeing that the system is fully connected. |   **Small World Property of a network:**  A small-world network is a type of graph in which most nodes can be reached from every other node by a small number of hops or steps. Human social networks, for example, famously connect any two people on Earth - or any player to Deiago Maradona - in six steps or less. This small world property has been observed for many real networks.  To examine if there is any ‘Small World’ property in a network, one can follow Watts & Strogatz’s method [1]. According to them, a network has the small world property if C >> Cr and L ≥ Lr. Here, Cr and Lr are respectively the clustering coefficient and characteristic path length for the corresponding random network having same number of nodes and edges.  For a random network having N number of nodes with average degree < k >, the characteristic path length (Lr) and the clustering coefficient (Cr) can be calculated using the expressions Lr ≈ lnN ln and Cr ≈ N given in [1].  Here we define the ratio p = C /Cr (12)  and the ratio q = L /Lr (13)  if p >> 1 and q ≈ 1 then we can say the network has ‘Small World’ property.  **Closeness Centrality**  Closeness centrality is used to find central vertices. It gives higher values to more central vertices. Closeness centrality of a node x, is denoted by Cres(x) and is calculated as follows  Cres(x) = N − 1 P yU,y6=x d(x, y)  where d(x, y) is the geodesic distance between node x and node y. U is the set of all nodes and N is the number of nodes in the network. The closeness value is therefore the inverse of the average distance between x and other nodes ( ¯d) i.e., Cres(x) = 1/ ¯d  **Residue Centrality**  The residue centrality is calculated using the changes of the characteristic path length under removal of node k with its links  ∆Lk = |L − Lrem,k|  where L is the characteristic path length given by Equation 9 with Lrem,k represents the characteristic path length after the removal of node k and corresponding links from the network. The statistically significant central nodes are evaluated using the Z-score values of the node centrality defined as  Zk = ∆Lk − ∆L σ  where, ∆Lk is the change of characteristic path length under removal of node k; ∆L is the change of characteristic path length under node removal averaged over all nodes in the network, σ is the corresponding standard deviation.  **Literature survey**  **A Statistical Framework for Streaming Graph Analysis [1]**  As we know that at present Social Networking is playing a vital role in every one’s life. Having a social networking profile allows the users to make and maintain business and as well as personal connections. And also, these profiles allow users to post anything and everything to the virtual world and can be seen by anybody.  The main advantage of having a social networking profile is it allows more chances for exposure and opportunities for marketing individual and his/her work. Twitter is one of the important social networking sites which allow us to make connections with others around the world who we don’t even know. However everyday lots of connections are going to be made and lot of data is going to share between the individuals. Twitter recorded over 13,000 tweets per second at its peak and also recently revealed that the service receives 400 million tweets per day on average. All these data should be securely stored and handled so that the individuals can use the twitter without any obstacles. But handling huge amounts of new data is not an easy task, they should be dynamically handled. This is where Graph topologies and analysis comes into picture.  In this paper the authors took a stream of Twitter posts (“Tweets”) from the time surrounding the land fall of Hurricane Sandy, a tropical storm that hit the Northern Atlantic coast of the United States, and formed a temporal social network. They computed the graph metrics, including betweenness centrality, in a streaming manner for each batch of new edges arising in the network.  **Temporal nature of a social network**  Authorsproposed a new methodology for gaining insight into the temporal aspects of social networks. They said that a solid foundation of analytical techniques is required in order to develop higher-level, large-scale data analysis methods for classification, prediction, and anomaly detection. The proposed techniques avoid the difficulty of predicting the new connections when already previous connections were given in a graph by modeling statistics computed from the graph over time. They demonstrated these techniques using a collection of Twitter posts related to Hurricane Sandy. With the available social network dataset with 1.2 million edges they studied the temporal nature of betweenness centrality and clustering coefficients while producing multiple visualizations of a network and also, they successfully detected the vertices whose triangle forming behavior is exceptional. The format of a Twitter post makes this information accessible. The authors mentioned about the following important aspects like **Related work**  **Graph kernels and statistics** which builds a data structure or an index on a graph and vertex statistic is nothing but a function from the vertex set to the real numbers that depends on the edge set and any information contained by the edges, i.e. edge weight. The data for each statistic can be stored as an |V|×|T| array, which is indexed by vertex set V and time steps T = t1,t2,...,tT. These dense matrices are susceptible to parallel processing using techniques from high performance linear algebra  **Global views of the data**  At the time, there was a great concern about the rapid development of Hurricane Sandy. Weather prediction gave more than one week of advanced notice and enabled the authors to build a tool chain to observe and monitor information regarding the storm on a social network from before the hurricane made land fall through the first weeks of the recovery effort**.** In order to focus the capture around the hurricane, they selected a set of hashtags (user-created metadata identifying a particular topic embedded within an individual Twitter post) that they identified relevant to the hurricane. And then they used a third-party Twitter client to combine our hashtags into a single stream containing a Java Script Object Notation (JSON) representation of the Tweets along with any media and geo-location data contained in the Tweets. They processed nearly 1.4 million public Twitter posts into an edge list starting from the day before the storm made landfall. This edge list also includes cases where a user "retweeted" or reposted another user’s post, because retweets mention the author of the original Tweet similar to a citation.  The dataset included over 1,238,109 mentions from 662,575 unique users. Then they constructed a graph from this file in which each username is represented as a vertex. The file contains a set of tuples containing two usernames which are used to create the edges in the graph. As the graph grows, the memory will eventually become exhausted, requiring edges to be deleted before new edge insertions can take place. At that point of time they do not consider this scenario, but proposed a framework by which they can analyze the graph in motion at a given moment in time.  **Betweenness Centrality**  In general centrality metrics on static graphs provide an algorithmic way to measure the relative importance of a vertex with respect to information flow through the graph. Higher centrality values generally indicate greater importance or influence. Betweenness centrality is a specific metric that is based on the fraction of shortest paths on which each vertex lies.  For this data set at time 98 they found that the right half of the distribution of logarithm of betweenness centrality is exponential with location 5.715 and λ = 1.205 Since betweenness centrality estimates are more accurate for the high centrality vertices, they focused their analysis on the vertices whose centrality is larger than the median. The cumulative distribution function (CDF) is 1−exp[−λx]. The above figure (Fig.1.) shows both the empirical CDF and the modeled CDF for the log (betweenness centrality). It is apparent in the figure that the exponential distribution is a good fit for the right tail.    **Observations**   1. **Observing a sample of vertices**     In the above figure (Fig. 2.), they traced the value of betweenness centrality for a selection of vertices over time. In this figure each series is analogous to a seismograph, depicting the fluctuations in centrality over time for a particular vertex. It is clear that there is a significant amount of activity for each vertex. Such a longitudinal study of vertices can be performed for any metric that can be devised for graphs.   1. **Analysis of derivatives**   The derivatives tracking of a statistic can provide us insight into changes that are occurring in a graph in real time.    They defined the (discrete) derivative of a vertex statistic for a vertex v at time t using the following equation where b(t) is the number of edges inserted during batch t.    The above figure (Fig.3.) shows the derivative of logarithm of betweenness centrality. These traces indicate that changes in the betweenness centrality of a vertex are larger and more volatile at the beginning of the observation and decrease in magnitude over time. The reason for taking logs before differentiation is that it effectively normalizes the derivative by the value for that vertex.   1. **Correlation**   In order to predict statistic values into the future based on current and past history, then we must identify a pattern in the temporal relationship that can be exploited. For any statistic, we can look at the Pearson correlation between the values at time t and time t+k for various values of k and quantify the strength of a linear relationship. However, this is not a method to predict statistic values into the future. Instead this quantifies the success of linear regression. They explained this using the local clustering coefficient metric.    Local Clustering of a vertex v is the number of triangles centered at v divided by degree(v)(degree(v)−1). The distance from each vertex to that best-fit surface could be used as a score for each vertex. The correlation function ρf(t,t+k) is a measure of how well a linear surface fits the data. Let ρf(t,t + k) denote the correlation between f(v) measured at time t and f(v) measured at time t+k. For this graph, ρf(t,t + k) is increasing in t and decreasing in k. There is a linear decay in the Pearson correlation coefficient as the gap size k increases. By fixing the batch size to a constant b, and the initial graph size to 0, they obtained the following equations for the relative impact of the batch at time t after a gap of k batches.    For a fixed time t, as the gap k grows, the relative impact of those batches grows. The above figure (Fig.4.) shows the correlation in clustering coefficient for the graph under consideration. The curves shown are ρf(t,t+k) where the series labels indicate t and the horizontal axis indicates the time of the second measurement which is t + k. The dashed lines are the best linear fits for each series. Since moving to the right decreases the correlation and moving to increasing series increases the correlation, the model is validated.   1. **Multivariate methods for Outlier Detection**   As we know anomaly detection is a huge problem, the authors focused on outlier detection, which is a more well-defined problem. The outliers of a data set are the points that appear in the lowest density region of the data set. One method for finding outliers is to assume that the data are multivariate Gaussian and use a robust estimate of mean and covariance which is known as the elliptic envelope.  As we have been computing the triangle counts and local clustering coefficient for each vertex in an on-line fashion, each vertex has a time series. This time series can be summarized by computing moments. They extracted the mean and variance of the original local clustering coefficient series. This is an embedding of the vertices into a real vector space that captures both topological information and the temporal changes to the network. This embedding can be used for any data mining task. Here they used the outlier detection to illustrate the usefulness of this embedding. Once the required features are extracted, the vertices can be displayed in a scatter plot matrix which shows the distribution of the data for each pair of features.  **Conclusion**  Detecting anomalous activity in a network is a key capability for streaming graph analysis. The paper mainly focused on using the vertex statistics (and their derivatives) to define vertex behavior, and then use outlier detection in the traditional way to detect vertices whose behavior differs significantly from the majority. The authors demonstrated this approach on a Twitter corpus by using a one class support vector machine where the behavior of interest is formation of triangles. This approach finds a partition of the vertices such that the inliers are tightly clustered and the outliers are diffuse.  **Screenshots**  <please refer next page>    Figure 1. Cumulative distribution function of log of betweenness centrality. A Experimental and empirical data    Figure 2.Histogram of log of Betweenness Centrality versus frequency    Figure 3. Correlation decay by Pearson method (dashed line – model, solid line – empirical)    Figure 4. Outlier detection by performing 'Analysis of derivatives' |

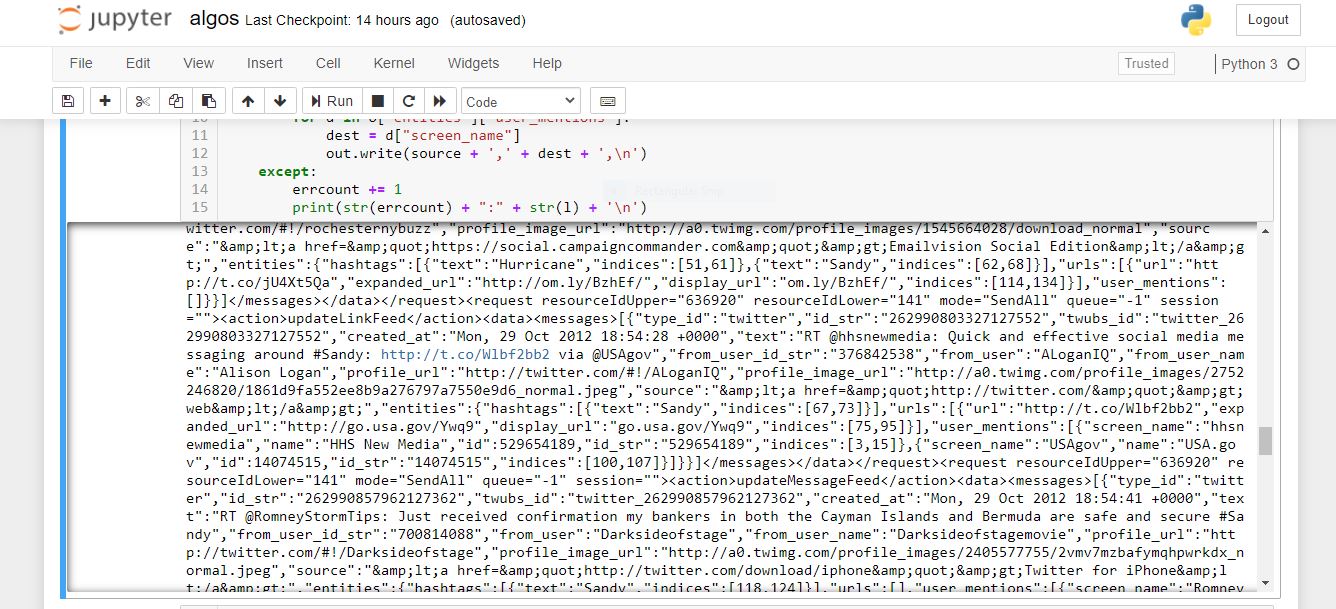
Screenshot of output:

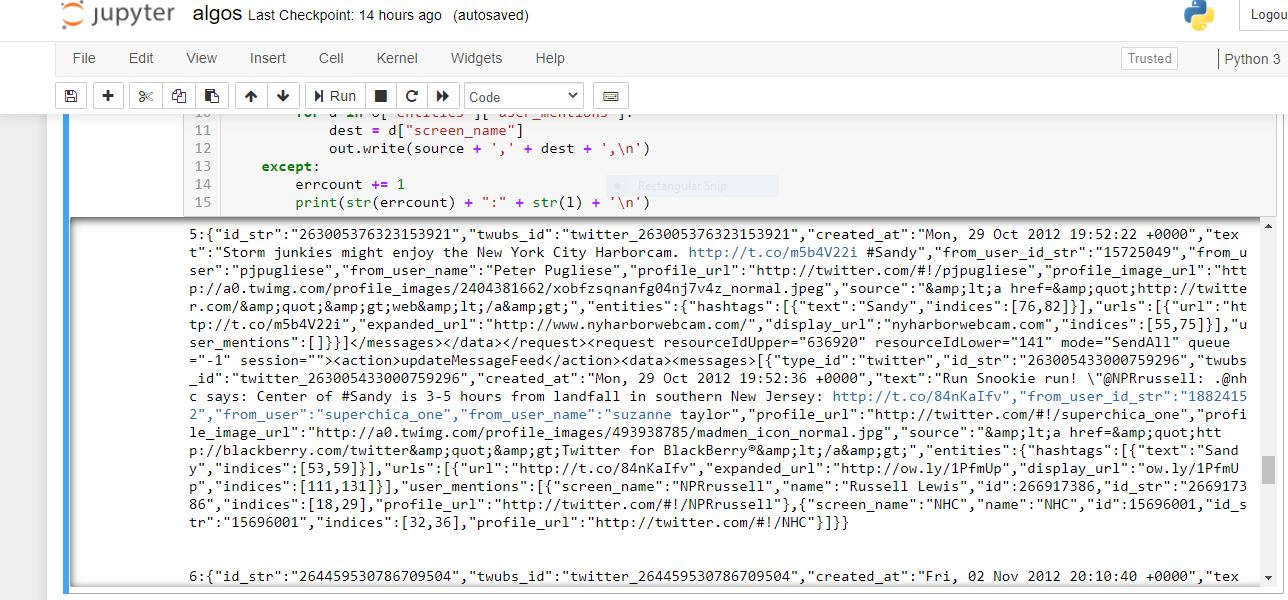


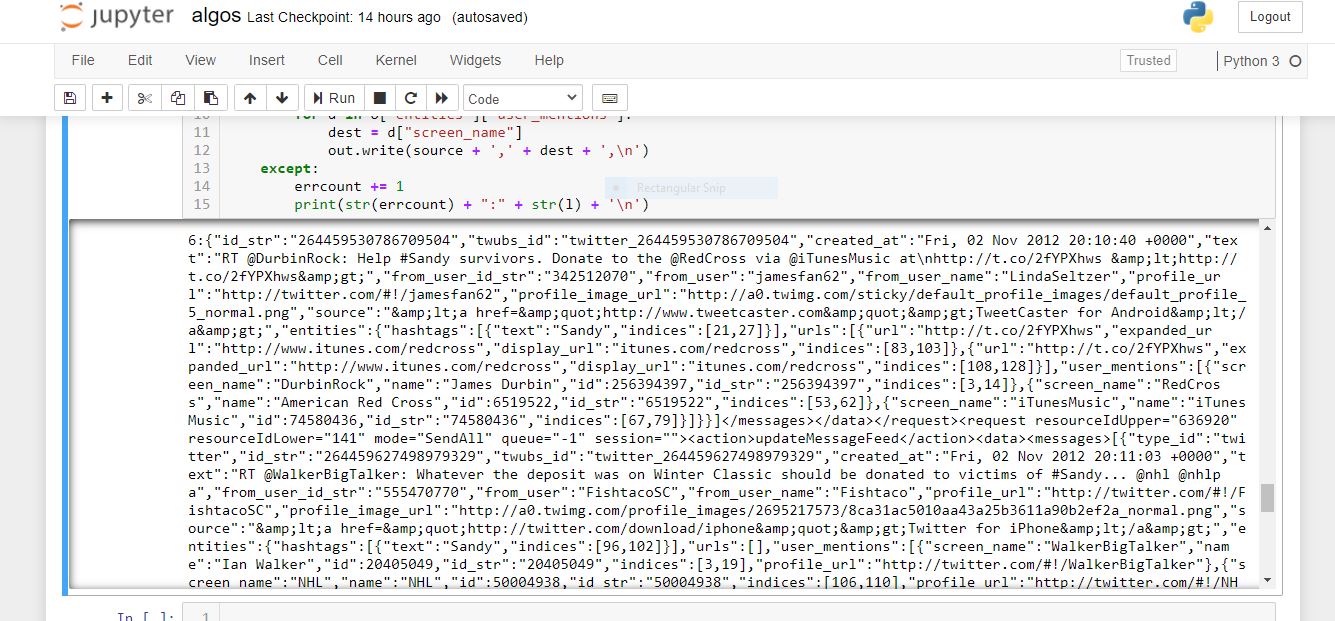


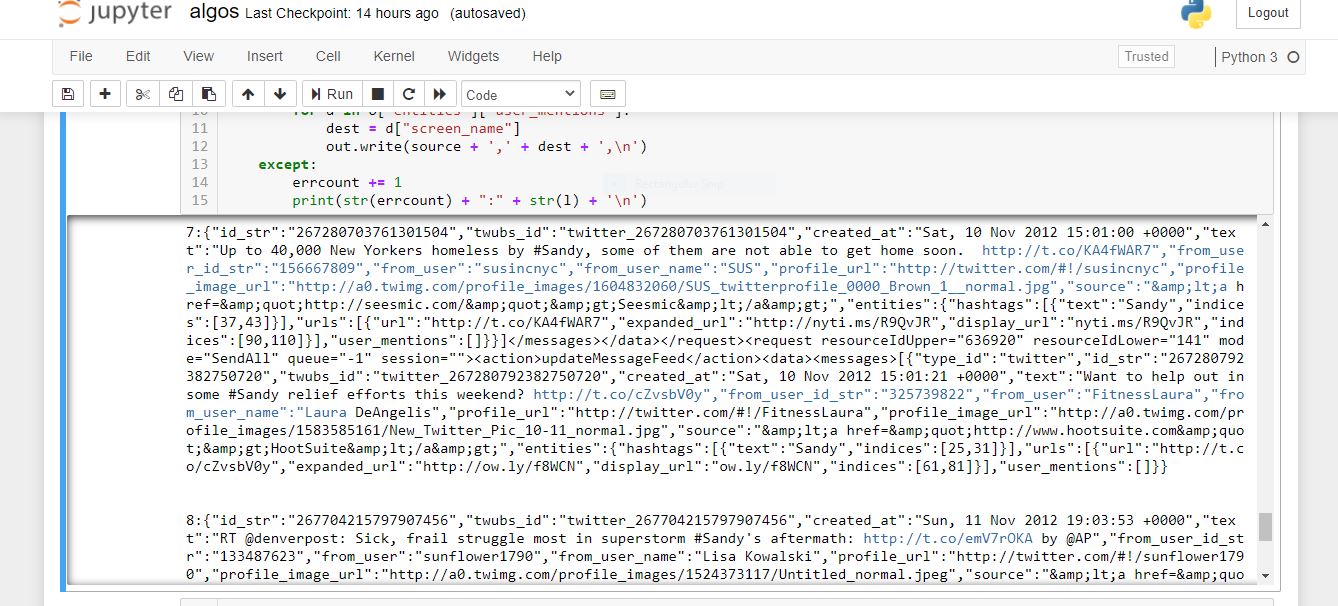


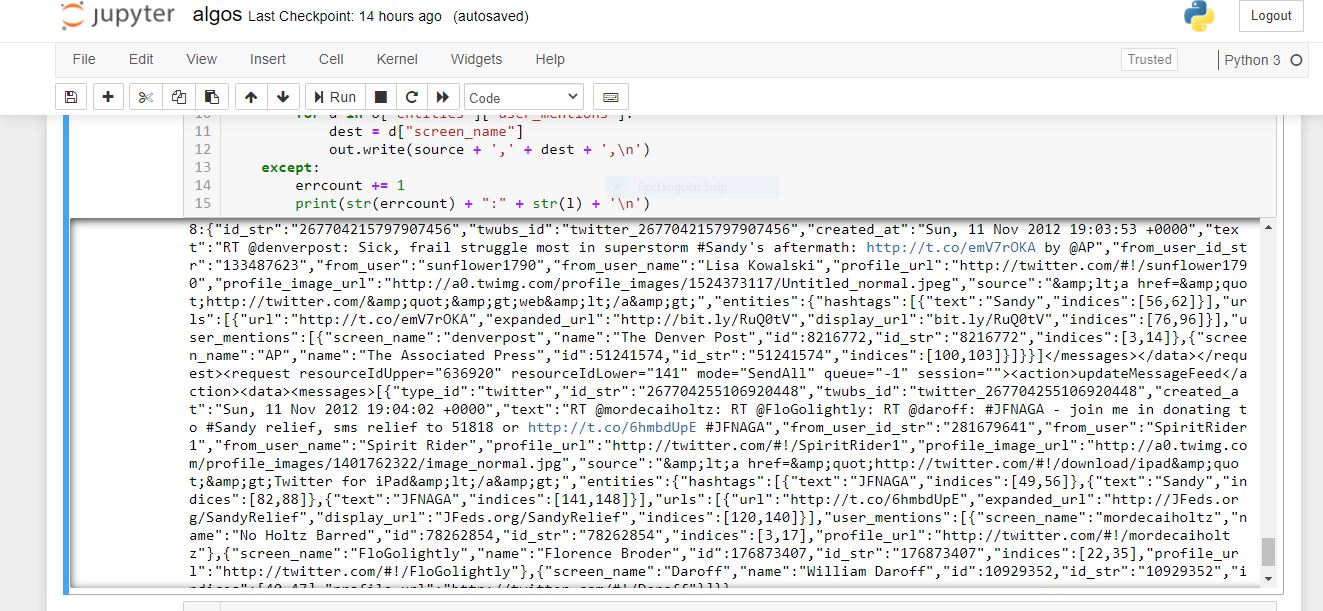












|  |
| --- |
| **References**  [1] - J. Fairbanks, D. Ediger, R. McColl, D. A. Bader and E. Gilbert, "A statistical framework for streaming graph analysis," 2013 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM 2013), Niagara Falls, ON, 2013, pp. 341-347. doi: 10.1145/2492517.2492620 URL: <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=6785729&isnumber=6785655>  [2] - https://en.wikipedia.org/wiki/Network\_topology  [3] - <https://uncc.instructure.com/courses/109743/files/6248767?module_item_id=1905643>  [4] - https://www.geeksforgeeks.org/applications-of-graph-data-structure/ |