Using Natural Language Processing to Assist With Internet Outage Detection

Max G. Sydow

Western Governors University

Table of Contents

Proposal Overview

When an internet service provider experiences an outage, customers notice, and will call or post to social media. It stands to reason that a spike in calls or posts may correlate with an actual outage, but what if more information can be gleaned from the content of those posts or calls? I would like to propose a method for the content of customer posts to be used for an ISP to possibly identify the cause of an outage. First, this involves extracting key words from content, and summarizing the frequency of those words. Times for which certain key words appear more frequent can be compared to times that actual outages of certain types have occurred. If a correlation exists between these key word spikes and known outage types, an ISP can use that information to guide resource allocation for efforts to further diagnose and resolve the issue.

An API can be used to extract content from Twitter and other social media platforms. Twitter posts contain only text and is limited to 280 characters, whereas Facebook posts can contain pictures, video, and other forms of content. Other social media platforms may not be as popular or relevant for a customer to communicate with their service provider. Twitter's APIs are also very accessible, so the scope of this project will focus on extracting data only from it. Other platforms certainly could be used, but those efforts may be more appropriate for a separate, similar project. Python has packages that can make sorting and cleaning the content easier, so that a more skeletal dataset can be checked for key words. Machine learning can then be applied to train and test for groupings of other relevant content pertaining to outage categories. Graphical displays of key word spikes can be made available on a dashboard, along with outage cause correlations. Such a graphical display can take the form of histograms made with Tableau. Further automation can occur when spikes from posts indicate a possible outage to send alerts to relevant departments involved with diagnoses and resolution. This automation can be triggered by operating system scripts.

Often times an internet outage is not noticed by an internet service provider (ISP) until a critical number of customer complaints and reports are received. These days many people have grown to rely on the internet in their homes and businesses. When that connection is down it can affect enjoying a streamed movie, a doorstep security camera feed, the ability of a child to finish their homework, and even loss of revenue due to inability of credit card processing. There are many avenues available now for a customer to alert their ISP when they are experiencing an issue. Sometimes hold times feel too long when calling, and many people have turned to email, and social media to air their concerns and complaints. Many ISPs have social media departments comprised of customer service and technical support representatives who assist customers via the platforms they use to communicate. The most popular being Facebook, Twitter, and to a lesser extent Google+ and Instagram. Other specialists may also communicate directly with customers via email. Some providers may have forums on their websites where customers can write posts on their issues. Other general sites such as DownDetector have forum sections where people can leave outage related comments as well. It would seem that with so many digital platforms being used by customers that a plethora of data could be mined to find insights on where an outage may be occurring and perhaps even the cause.

The backbones of the internet rely on complex networks of hardware and software. Fiber optic and copper cabling connect central offices consisting of thousands of routers, servers, and other node devices. Protocols such as TCP/IP and BGP guide the rules for data packet flow through the physical components, while servers are specifically configured to control certain types of data such as email and credit card transactions. It is impossible for an ISP to keep a constant eye on each inch of cabling to spot damage, especially since much of that cabling is buried under ground. Packet traffic can be monitored by syslogs, routing tables, and other software, but often times those tools are accessible via direct connection to specific routing devices. Teams of network administrators and engineers use those logs, but when there are millions of them each with millions of lines of output it becomes impossible for humans to constantly monitor those to find traffic jams or breakdowns. Even errors with specific servers can be difficult to spot immediately by the administrators who manage them. Basically, there are so many factors that could cause an outage that most of the time resolving one is a re-active task as opposed to pro-active.

Internet outages can be attributed to a reasonable collection of common causes. A survey of 315 network professionals at mid to large size enterprises was conducted by Veriflow (Bednarz, 2016). Another blog post on the Fastmetrics forum titled *What Can You Do When an Internet Outage Occurs?* describes the top 5 causes of internet outages. Another blog post from Geek Speak (vinod.mohan, 2013) identified the top 10 reasons for network downtime. By comparing these 3 sources, a list of most common outage categories can be made. They are, in no particular order:

1. Human error
2. Electrical outage
3. Hardware damage
4. Software issue
5. Network congestion
6. Cable damage
7. Security attack
8. Natural disaster

Most people cannot tell if there is a software issue, routing table conflict, coding error in a server file, or a DDoS attack. Extreme weather, cable or physical hub damage near one's home or business, or even shoddy work seems more likely to be observed. On the other hand, network engineers do subscribe to internet services in their homes, and other knowledgeable customers may be able to make better guesses on why their internet services are currently down. When customers post comments about their service issues the content of those messages may vary widely in level of detail. Much like how a medical doctor can use a patient's description of their symptoms to aid in diagnoses, customers descriptions of their issues can aid in determining their causes.

Work has been done to examine word usage patterns from lists of emails, social media content, and even audio recordings of calls. Natural language processing (NLP) is a form of machine learning that can be applied to find patterns in words and phrases and draw conclusions. If there is a spike of social media posts from customers in a certain area containing content describing stormy conditions and no wi-fi on multiple devices, there may be cause to look for a downed overhead wire, or power outage at a central office. The frequency of key words such as "stormy", and "wi-fi" can be extracted from these posts. Other such trends in content can be examined during time intervals in which past outages have occurred. If spikes in words in the above example have been observed at the time of a past known issue that was attributed to a downed line, then a justification for correlation of message content and outage cause can be made. Here in lies the power of using machine learning in the form of NLP to help ISPs engage pro-active outage response tactics.

Implementation Plan

A small team of analysts and/or data scientists, and developers need to be chosen to undertake the work for this project. The overall goal is to use Tweets to provide alerts for possible outages. This initial task of extract, transform, and load (ETL) from past Tweets can be performed by analysts. Some cleaning of the content extracted is needed to better prepare the collection of words and phrases obtained for natural language processing. Once NLP is applied a set of word groupings can be obtained and comparison of the timing in higher frequency occurrences of those groupings can be made with data from the ISPs existing outage reporting system. Any correlation between existing outages and the occurrence of observed word groupings would indicate a possible outage. The data from ETL on Tweets and past data can be joined into the same database. A threshold of word grouping occurrences can be established as criteria for a possible outage. These tasks can be performed by analysts or data scientists. The word groupings and their frequencies can then be visualized in histogram form and added to the existing outage reporting system. The histograms can be prepared using Tableau by analysts, while adding it to the existing system can be allocated to developers. This dashboard will need continual updating as new Tweets come in.

The same ETL and cleaning code can be applied to future Tweets and need to be periodically ran. The NLP code also needs to be run each time extraction is made to make sure word groupings remain consistent. At this point the work of analysts and data scientists is complete for the purposes of this project; developers can write the scripts to periodically execute these tasks. Developers would also be tasked to write the scripts to update the visual dashboard and trigger email alerts.

Review of Other Work

In order to extract key words and phrases the content of Tweets need to be parsed and processed. Having to filter each word of each post in order to mine key terms could be very time consuming. Storing all content may also not the best use of memory for a database. In *Internet Outages, the Eyewitness Accounts: Analysis of the Outages Mailing List*, techniques from text mining and NLP to perform data preprocessing in a slightly different context were described. This white paper examines key words from a publicly available internet outages mailing list to categorize type of outage and entity involved.

One of the first things that the authors of this paper did to narrow down content was to remove stop-words. Stop-words include articles, prepositions, and pronouns. The SMART informational retrieval system, which is an early NLP system developed at Cornell University in the 1960s, was used to identify such stop-words (Banerjee et al., p. 4). This functionality can be performed by Python packages, which will be discussed later. Punctuation was also removed.

Lemmatization is the process of grouping together different forms of words. The verb "to talk" can be grouped along with "talk", "talking", "talked", etc. Again, the authors used an older tool that can be supplanted by Python modules for this function. The Natural Language Tool Kit, or nltk, package can be imported into Python code to perform lemmatization. Names of people, and organization names were also removed. I would argue that organization names such as "level 2" or "engineering" not be removed, as a knowledgeable customer's expression of intuition may be a useful indicator.

Another group whose paper is closely related to the one described above describes data processing a little differently. In *Network Outages Analysis and Real-Time Prediction,* some more content extraction criterion was discussed. Links to other websites and emails were filtered out, as were abbreviated words such as ICS, and ISP. Apparently, output from Traceroutes were found in these emails, and they were also removed. I wouldn't expect to see those in customer Tweets, but if they do appear, they would be more useful to networking experts than as a means of classifying language describing services being down.

Applying natural language processing on Tweets was discussed in an article on the towardsdatascience.com site titled *Can We Use Social Media to Locate Legitimate Power Outages?* Although the purpose of the article was focused on identifying electrical power outages, many of the same techniques can be applied to identifying internet outages. The author discusses using Twitter's API and TwitterScraper to extract semantic content.

As in the previous projects discussed, common stop-words were removed before looking at frequency of relevant words. Tweets from several of the largest US cities were scraped covering a time span of 5 years. The author indicates a preference for using TwitterScraper as opposed to Twitter's API. While the API includes geolocations, it only provides one month of historical data. For ongoing analysis Twitter's API may be used.

A set of key words and phrases such as "outage", "power failure", "electricity out" etc. were used to identify Tweets in targeted cities from which to further examine. Hill only mentions spending time exploring how people talk about power online. There would seem to be a great deal of subjectivity here, as there likely will be when describing internet outages. One would have to trust that if someone Tweets that they are experiencing a "power failure", then there exists a reason beyond the premise for cause – i.e. an outage. For the purposes of this project, a basic common sensical list can be comprised. Further research on variations of vocabulary used by customers and networking professionals may help to serve as future goals to enhance the efficacy of possible side projects which may grow from this one.

The most frequently used words and 2-3-word long phrases were extracted from the Tweet set and composed into a data frame. Some further cleaning was performed on the data frames before modelling. The Word2Vec model was used "because of the way it focuses on the relationship of words and gives weight to that value". (Hill, 2016, para. 12). This model can be trained to learn conceptual relationships between words. Since Word2Vec involves computations on higher dimensional vector spaces, a t-SNE model was used for dimensional reduction for purposes of visualization in 2-dimensions. The author then uses a cosine similarity measure to numerically gauge how closely the most frequent words and phrases correspond to targeted key words. See Appendix A for an example of a t-SNE plot. The higher the score the more related the other words and phrases are to the existence of an actual outage.

While the mathematical details of how Word2Vec, t-SNE, and cosine similarity are beyond the scope of this discussion, some conceptualization on how these algorithms work is warranted. The following summary is backed up by a Wiki titled *A Beginner's Guide to Word2Vec and Natural Word Embeddings*. The content of the cleaned Tweet data set can be thought of as a "bag of words". A frequency count of the occurrence of each word in this bag can be made by simply counting and sorting. Basically, a corpus, or collection of similar words is determined from the bag. Each word is associated with a vector, and probabilities are calculated on how closely related they are based on abstracted context rules.

The independent variable in a simple cosine function can be in the form of degrees. 2 linearly independent vectors in 2-dimensional space are orthogonal, or 90 degrees apart. Determining closeness of one vector to another can be thought of in terms of angular degree separation. The output of a simple cosine function ranges from 0, indicating overlap to 1, indicating 90-degree separation. Word relatedness can be measured this way in the context of Word2Vec using Cosine Similarity. Nicholson describes an example: "Sweden equals Sweden, while Norway has a cosine distance of 0.760124 from Sweden, the highest of any other country."

Once the frequency of the most common words appearing in Tweets that contain general terms related to internet outages are found, a corpus of other similar words can be found using the above algorithms. Python has modules that can be used, and the author of the power outage article provided a link to a GitHub repository including TwitterScraper, and NLP algorithm use. This code can be adapted and used in this context.

Relation of Artifacts to Project Development

Specific examples of an ISP undertaking a project like this could not be found, as ISPs keep their methods of operations unavailable to the public. Electricity and internet providers can both be thought of as utility providers, so the concepts of that work can be modified and applied in this context. The same methods Hill used for detecting power outages using Twitter can be directly applied for ISP outages. The initial criteria to scrape Tweets and begin to form related word groupings will, of course, be different. Even though Banerjee et al. and Zhu et al. utilize an email list, the cleaning and lemmatization procedures discussed by can be applied to Tweet content. Nicholson's article helps us understand how we can start with a bag of words and end up with quantifiable measures of how similar other content is to the words in that bag. In general, the building blocks of this project can be observed from different contexts and modified to achieve the specific goals for an ISP.

Project Rationale

According to the article, *When the Internet Goes Down: Tracking Edge Outages at Scale*, smaller scale local outages are more scattered and difficult to detect. "These outages are typically not visible when studying the Internet's control plane (the routing information exchanged between networks). Thus, detecting these potentially small events is similar to finding a needle in a haystack." (Richter, 2018) I've witnessed this effect firsthand while working as a customer facing technical support representative for an ISP. All of the sudden a spike in calls will occur from customers in the same vicinity, reporting similar issues. No common cause or outage report has been issued to provide further information. So, after typical exhaustive troubleshooting with individuals an internal ticket will be escalated for further testing. The time it takes from an initial spike in calls to diagnosing and declaring an outage can vary. In my own experience I've seen that time gap range from the order of several minutes up to several days.

These days most ISPs have social media teams who respond to customer complaints via Twitter, Facebook, and other platforms. But like traditional call center based customer support the efforts on social media response is reactionary and attended to on a case by case basis. Individual posts and threads are responded to and assistance is provided based on message content. I'm not advocating doing away with reactionary responses to Tweets, because issues will still arise that can be resolved through guided troubleshooting customer equipment in the home that is not outage related. Before a social media representative has a chance to even read the content of one of the nearly constant influx of new Tweets it's content can be automatically extracted and analyzed.

Current Project Environment

The current state of an ISP for which this project would apply is one that has not yet incorporated behind the scenes automated analysis of social media content. Specifically, there is a team of technical support representatives who respond to customer issues via Twitter. The ISP does not store content of these posts, but representatives update case logs of interaction summaries that are stored on an existing database. Customer identifying information such as name, username, and location (if present), are included in these summaries. Past Tweets can be scraped by username or mention of the ISP's name, and from here NLP can be performed. A database that contains details on past outages exists, including cause of issue, outage duration, and location identifiers. Results from extracted and analyzed content can then be compared to known issues on record to find even more realistic correlations.

For example, it may have been found that there was a spike in the term "Wi-Fi" for a certain 10-hour period. NLP indicates this spike is likely related to an internet outage. Cross referencing existing data confirms there was indeed an outage caused by a misconfigured routing table that serves that specific area with a duration of 12 hours that overlapped the time frame in which that key word spike appeared in extracted Tweet content. This would confirm the NLP output as an accurate predictor of an actual outage for this particular instance. Of course, instances that do not provide correlation may be found. Different NLP algorithms can be experimented with to find which gives the best correlations.

Trends on outages from existing data have been analyzed, and there are systems in place that showcase such statistics. The outage record and reporting system is made available to relevant work groups including management, network operations, and technical customer service. The content of these reports is accessible via a secured URL, and the database that holds its content is capable of exporting CSV's. Email alerts on newly created outage reports are also automatically sent to specific groups involved with resolution efforts. Restoral status is also automatically relayed to customers via web-portal, email, and SMS.

An additional database will be needed to store sorted content extracted from past Tweets. An adjacent table in that data base can be created to hold newly extracted content. This may require additional servers, or at least more space on existing DB servers. Workstations will be needed to perform ETL (extract transform and load) tasks, data cleaning, and NLP processing. The CSV's from these databases can be imported to workstations used by analysts and data scientists via Python and converted to data frames. The bag of words in the cleaned data frames can then be split into training and testing feature sets for use with NLP algorithms.

Meanwhile a web scraper can be used to extract meaningful data from the existing outage reporting system. Another table in the Tweet content DB can be added linked via time-stamp. These tables can be unioned and iterated over to find co-instances of key word spikes and actual outages. Further work by data scientists and analysts can be performed at this stage to determine which NLP algorithms provide the best correlation with actual outages. Ongoing efforts in this direction is likely a different project in and of itself, but one or 2 algorithms can be used to start with.

Active updates from incoming Tweets need to periodically be uploaded into the DB, and the Python scripts used for cleaning and NLP can be automatically triggered. Operating system batch scripts can trigger these ETL and analysis functions. Criteria from NLP performed on past data can be used to trigger alerts of possible occurring outages. These alerts can be displayed graphically on a dashboard using Tableau. Keyword clusters and their corresponding frequency of appearance can be displayed in histogram form and color coded to indicate likelihood of outage. The display itself can be incorporated as another tab or section of the existing outage reporting system. Functionality for triggering email alerts to appropriate departments can then be added to existing automation.

Methodology

This project can be thought of as the first phase of much larger endeavor. Its focus is only on analyzing Tweets, and the initial list of words used to filter them might be fairly basic. Ongoing processing of other social media platforms and expanding or altering that initial word list may be pathways to related projects that can enhance the overall goal of better predicting outages. A relatively small team of analysts, data scientists and developers can accomplish this first phase. The Agile framework utilizes self-organizing and cross-functional teams in collaboration to develop a product. In the following section, a set of objectives are defined in a way that each one can be accomplished in a finite time frame. Two flavors of Agile may be applicable here: Scrum and Kanban.

Scrum breaks a project down into sprints, which involve team members working in a focused manner on a specific objective. Scrum meetings are scheduled to communicate progress on achieving benchmarks. Little scope creep or deviation is allowed, which keeps teams focused on their goals defined by the sprints. The path from extracting Tweets to presenting a functional dashboard can be broken down into manageable tasks. Another reason I chose to limit the scope of this project to Twitter and using a basic list of words as extraction criteria is to limit scope creep.

It's not hard to imagine ongoing brainstorming sessions for different word lists. The terms "speed" and "bandwidth" are often used interchangeably when it comes to internet speed. One list including one of these words may produce a different set of extracted Tweets than the other, and NLP will likely group these terms together anyway. Questions like whether to use one or the other, or both may arise and could be a cause for debate. The output of NLP may also vary according to which extracted Tweets are used, which would snowball into a different result of outage correlation. I think it's best to start with a specific set of extraction words and go from there; verifying some correlation and making the results available are the most crucial parts of getting this project initially put into place. Ongoing data science heavy projects may be spawned from these concerns to make predictions more accurate. On his website, data scientist Chang Hsin Lee writes on the disadvantages of using Scrum for data science projects. The Kanban flavor of Agile may be more appropriate for one of those offshoot projects as it is more focused on task than timeline.

There will likely be other challenges when wrangling data from other social media platforms. API's may allow for extraction of different fields, timestamp and location formats may vary or be limited. Facebook allows for more than the 280 characters that Twitter is limited to, and storage may become an important consideration when extracting from it. Applying the same goals to other platforms would likely follow a similar project outline that can also use the Scrum methodology. If all social media platforms were worked together, the ETL phase of using Facebook may end up taking much longer than with Twitter. To incorporate other platforms into this single project could give rise to further deviation of sprint goals. Narrowing the focus of using just one platform to start with allows for efficient application of teams achieving timely defined objectives, which is Scrum.

Project Goals, Objectives, and Deliverables

Goals, Objectives, and Deliverables Description

The following is a list of objectives required for this project:

1. Scrape data from Tweets by username going back 5 yrs.
2. Clean data – remove stop words, etc.
3. Load cleaned data into Tweet database
4. Prepare data frames and conduct NLP
5. Extract data from existing outage reporting system going back 5 yrs.
6. Link and union existing DB to Tweet DB
7. Perform correlation study
8. Branch off to side project of finding best NLP algorithm
9. Determine criteria for outage likeliness based on NLP output
10. Write scripts to trigger ongoing periodic ETL and analysis for incoming Tweets
11. Create visual dashboard of key words found in Tweets with outage likelihood indicators
12. Update functionality of outage reporting system to include alerts based on dashboard

The first phase is to perform extract-transform-load (ETL) on Tweets going five years back. This should provide plenty of data to draw from and compare to 5 years' worth of actual outage information in an ISPs existing database. With the exception of newer internet of things (IoT) devices such as thermostats, and other appliances most of the same type of issues were also reported 5 years ago. These include issues with loading webpages on laptops or desktops, issues with streaming video, and smart-phone and tablet connectivity. Including criteria in filtering word lists to describe newer IoT issues can be postponed and included in one of the offshoot projects described earlier.

The Tweet ETL phase involves utilizing its APIs and applying the filtering word list to extract relevant Tweets. The content then needs to be cleaned to sift out stop words, and other spurious content, and a resulting Python data frame can be constructed. A list such as:

{internet, wi-fi, buffering, webpage, slow speed, can't connect, timeout, router, cable, damage, power, storm}

can be used as a matching criterion to extract relevant Tweets. Any posts containing these words will be uploaded and cleaned to form the data frames.

A second and simultaneous ETL phase can be performed on the ISPs existing outage database to grab outage causes, durations, and locations.

NLP can then be applied to the data frame to find and group clusters of related content. Python's Word2Vec can be used here and like in the article on power outages mentioned above, t-SNE plots can be made to visualize the strength of relatedness of content to the original key word list describing outages. In such a plot words or phrases are represented as points on a 2-dimensional plane, and clustered together and often color coded according to the groupings found from Word2Vec processing. In this case each cluster represents words pertaining to a certain type of outage. Ideally the clusters found will group into categories that come close to aligning with the typical categories identified in the overview section of this proposal.

Time stamps on Tweets containing words in each grouping can then be cross referenced with time data in the ISPs existing outage database. To start we can look at the time durations for a known outage, then look at the number of Tweets by word grouping for that time interval. For example, a high frequency of grouped words that indicate an internet outage caused by a storm according to NLP were found coincide with an actual storm related outage on record. Now, if more such coincidences were found that may imply that higher frequency of Tweet counts for a specific word grouping can be associated with an occurrence of an actual outage. There will be an average number of Tweet counts that correspond with the occurrence of an actual outage μ_1, and an average number that do not μ_2 . We want the first mean to be greater than the second. An independent samples T-test can allow for rejection of null hypothesis that the first mean is less than or equal to the second mean. The mean minus standard deviation of the corresponding group can give a threshold Tweet frequency count above which to indicate that an actual outage may be occurring.

Weak correlation may mean that a certain grouping isn't useful and should be omitted as an outage predictor. On the other hand, it could lead to using different original Tweet filtering word list criteria, or even a different NLP algorithm such as K-means clustering. Some adjusting should be allowed during this phase, but not so much to significantly prolong the duration of the project. If necessary, trying 5 lists of filtering words and 2 NLP algorithms should be allowed during this phase. If no strong correlation is found, then incorporate those efforts into a separate project. This choice is rather arbitrary but seems like a reasonable amount of allowable scope creep to not completely derail progress.

The same color coding from the t-SNE plots can be used to color code the frequencies of such word groupings in histogram form over certain time durations. Such a plot can be made using Tableau and added to the final dashboard which can be included in the ISPs existing outage reporting system. Scripts to periodically extract word groupings from ongoing Tweets can be made as well as update the dashboard histogram. When a certain frequency of associated words appears above the threshold described above then scripts to send emails notifying relevant work groups can be sent to alert for a possible type of outage. A summary explaining the possible occurrence of an outage can also be added to customer facing website and app as an enhanced version of existing issue tracker.

Goals, Objectives, and Deliverables Table

The following is a table summarizing each goal, it's deliverables, and objectives in achieving it (some goals have more objectives than others).

Goals	1 ETL on Tweets	2 ETL on existing outage DB	3 Perform NLP on Tweet data frame	4 Compare results from outage DB and NLP groupings	5 Dashboard	6 Ongoing monitoring and implementation
Objectives	Extract Tweets containing pre-defined outage related words using TwitterScraper	Extract outage cause, times, and location	Apply Word2Vec algorithm	For time intervals for actual outages, examine frequency of word groupings	Add visual summary from previous goal to existing outage reporting system	Scripts to continuously and periodically run ETL and NLP code on new Tweets
	Clean content – remove stop words, and other spurious content		Apply cosine similarity and t-SNE	Tabulate true and false correspondences		Scripts to alert workgroups if/when threshold is exceeded
				Perform T-tests to reject null hypothesis: $\mu_1 \leq \mu 2$		
				For correspondences that meet above criteria compute mean minus standard deviation		
Deliverables	Cleaned data frame of words and phrases to be processed by NLP	Table containing outage cause, times, and locations	Groupings of related words to outage causes, and t-SNE plot to visualize	Summary of word groupings that correspond to actual outages, and numerical threshold of group frequency	Visual dashboard with histogram of word group frequencies color coded by cause, and threshold indicated	Verification that dashboard from previous goal gets updated, and confirmation of received emails

Project Timeline with Milestones

Timeline and milestones can correspond with the objectives and deliverables outlined above. The Scrum project methodology is intended for focused projects to be completed in a relatively short time that can be measured in days to weeks. Planned Scrum meetings occur during the project and are typically scheduled when an objective is completed to present a deliverable. It is easy, then, to map milestones to the 6 numbered goals and deliverables above. This project can be estimated to be completed in a matter of days if 2 goals were worked each day. The last objective involves more ongoing monitoring of functionality, so if work is started on a Tuesday, we can allow for the remainder of a 5-day workweek to check for any bugs. That will also allow for some cushion if a milestone needs to bleed into another day to complete.

Milestone	Duration	Start Date	End Date
Goal 1: Cleaned Data Frame	½ Day	2/11/2020	2/11/2020
Goal 2: Table from existing outage DB	½ Day	2/11/2020	2/11/2020
Goal 3: Summary of word groupings from NLP	½ Day	2/12/2020	2/12/2020
Goal 4: T-test comparisons	½ Day	2/12/2020	2/12/2020
Goal 5: Dashboard with histogram summaries	½ Day	2/13/2020	2/13/2020
Goal 6: Automation of continued analysis: scripts to run ETL and NLP, and send emails	2 Days	2/13/2020	2/14/2020

Outcome

One indicator of success is how well the output of a particular NLP algorithm correlates with the existence of an actual outage. This is measured using the T-tests indicated above. However, it may turn out that some categories do not meet the criteria of rejecting the null hypothesis. Again, this may mean that a modified word list for original Tweet selection is needed, or a different NLP algorithm should be tried. For the groupings that pass the T-test and can be used as indication of possible outage can be continually monitored for consistent meeting of that criteria. In other words, the frequency at which an actual outage is confirmed corresponding to alerts based on NLP is one measure of effectiveness.

Ongoing monitoring of the performance of scripts to run ETL and NLP processing is needed to make sure they run at the times they are scheduled for. Scripts to update the dashboard also need to be monitored so that no update is missed. Receipt of emails should be confirmed as well, and feedback from relevant workgroups should be advised. When this project is implemented, it’s effectiveness it is imperative to know how often alerts actually lead to diagnosis of actual outages. Another key indicator is if diagnosis is found to be quicker than it was before.

If this method of outage indication using Tweets proves to be effective, then expansion to other forms of social media can be undertaken. These projects can follow a similar template of implementation as outlined in this proposal, and the same means of gauging effectiveness can be applied. In time, if effectiveness with other platforms is demonstrated then alerts can be sent to customers as well. It is probably wise to not do this right away so as not to give customers any false hopes. Responses from customer satisfaction surveys can be elicited, and yet other NLP projects can be spawned to gather input from social media on perception of this new feature available to them. Hopefully perception will become more positive, which may in turn increase customer retention, attract new customers, and increase and ISPs stock value.

References

Bednarz, Ann. (Nov 18, 2016). Top Reasons for Network Downtime.

Retrieved from https://www.networkworld.com/article/3142838/top-reasons-for-network-downtime.html

What Can You Do When an Internet Outage Occurs? (n.d.). *Fastmetrics*.

Retrieved from https://www.fastmetrics.com/blog/tech/internet-outage-what-to-do-now/

vinod.mohan (Aug. 29, 2013). Top 10 Reasons for Network Downtime. *Geek Speak.*

Retrieved from https://thwack.solarwinds.com/community/solarwinds-community/geek-speak/blog/2013/08/29/top-10-reasons-for-network-downtime

Banerjee, R., Razaphpahah, A., Chiang, L., Mishra, A.,, Sekar, V., Choi, Y., & Gill, P. (Nov 16, 2015). Internet Outages, the Eyewitness Accounts: Analysis of the Outages Mailing List.

Retrieved from https://users.ece.cmu.edu/~vsekar/papers/pam15_outages.pdf

Zhu, G., Wei-Ting, L., & Sun. Z. (n.d.). Network Outages Analysis and Real-Time Prediction

Retrieved from http://zhuguanyu.github.io/fundamental_of_network/documents/[2]Network_Outage_Analysis_and_Real-Time_Prediction.pdf

Hill, Jen. (Apr 29, 2019). Can We Use Social Media to Locate Legitimate Power Outages? *towardsdatascience.com*

Retrieved from https://towardsdatascience.com/can-we-use-social-media-to-locate-legitimate-power-outages-7b7409708447

Ferris, Patrick. (Aug 14, 2018). Learn TensorFlow, the Word2Vec model, and the TSNE algorithm using rock bands. *freeCodeCamp*

Retrieved from https://www.freecodecamp.org/news/learn-tensorflow-the-word2vec-model-and-the-tsne-algorithm-using-rock-bands-97c99b5dcb3a/

A Beginner's Guide to Word2Vec and Natural Word Embeddings. (n.d.) *pathmind A.I. Wiki*

Retrieved from https://pathmind.com/wiki/word2vec

Richter, Philipp. (Nov 30, 2018). When the Internet Goes Down: Tracking Edge Outages at Scale.

Retrieved from https://labs.ripe.net/Members/philipp_richter/when-the-internet-goes-down-tracking-edge-outages-at-scale

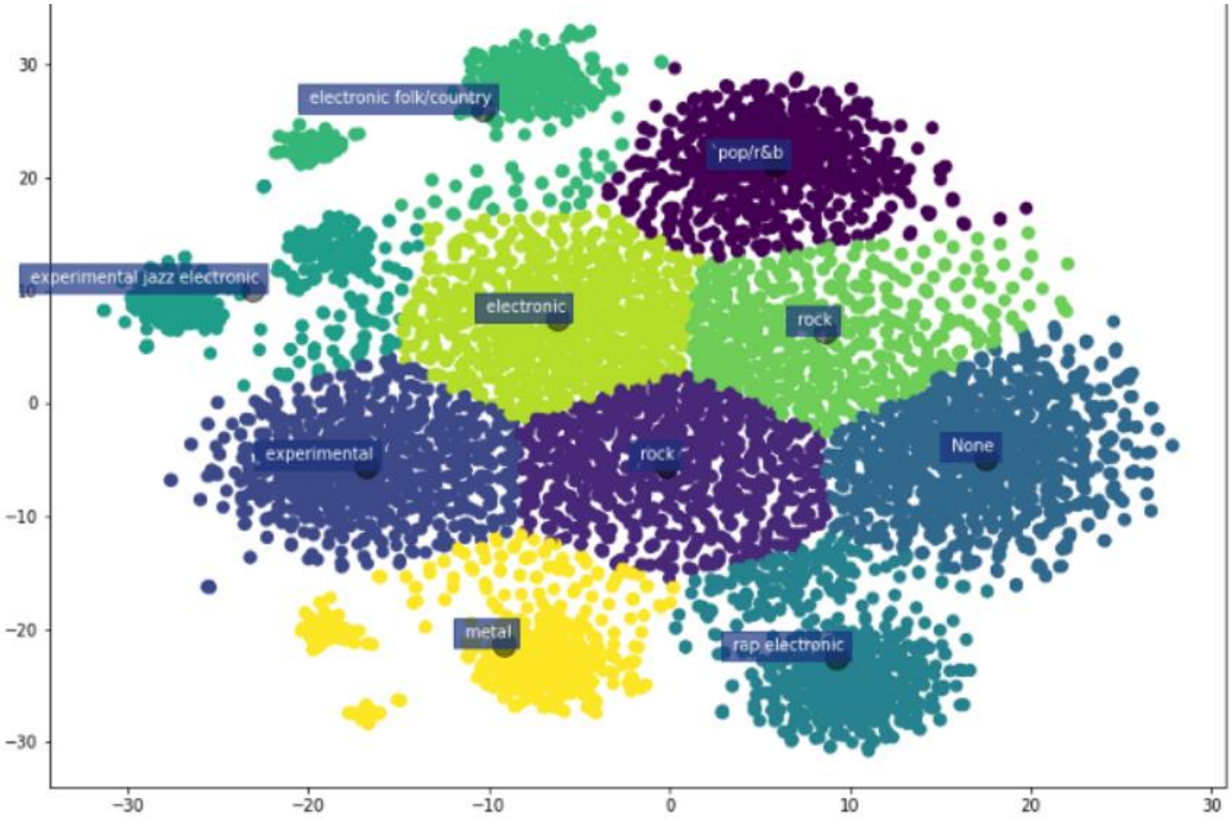
Hsin Lee, Chang. (n.d.) Agile in Data Science: Why My Scrum Doesn't Work? *changhsinlee.com*

Retrieved from https://changhsinlee.com/agile-ds-scrum-kanban/

Appendix A

t-SNE Plot

To better understand what a t-SNE plot is, refer to the image below, which was taken from a freeCodeCamp forum post by Ferris (2018). This particular plot groups words related to music genres, where words describing different genres are represented as points and grouped together in color coded clusters. The clustering for this internet outage project would group around different words such as “connection”, “slow speed”, etc. While the subject matter is different, I believe it is useful to include a visual example to portray what a t-SNE plot is.


electronic folk/country
pop/r&b
experimental jazz electronic
electronic
rock
experimental
rock
None
metal
rap electronic