California Camp Fire

The Outliers

Christina Chang, Jonathan Quach, Troy Lui

Table of Contents

Background

Project Goal

The California Camp Fires is a wildfire that started within the town of Paradise in Butte County. The fire lasted around 2.5 weeks, which started on November 8th, 2018 and was 100% contained on November 25th. From first glance, we wondered whether there are differences in digesting news from a social media, local, and national source. To supplement this, we used Twitter to represent the social media view, the Paradise Post to represent local news since the fire started in the town of Paradise, and Google News to represent national news since it encompasses multiple news sources. To note, Google News does contain Paradise Post articles; however, these articles were extracted to holistically represent our view of "national".

Hypothesis: Local news will approach coverage on the Camp Fire a more episodic perspective compared to national and social media news sources.

A news article can be viewed with an episodic or thematic perspective. An episodic view focuses more on the individual. Articles written with an episodic view include single events and individual interactions. On the other hand, a thematic view is more broad; they focus on general trends and the context or environment of an event.

We initially supported this hypothesis because we believed that the closer one is to the event, the more specific and personal articles pertaining to the event will get. In short, we believed that local articles would include news about local schools and people safety.

Follow-Up Project Goal

After delving into more exploratory data analysis, such as Word Clouds and frequency plots, we decided to make a follow-up hypothesis, which answered the question, "What kind of topics did news sources cover about the Camp Fire?" From this, we made a follow-up hypothesis.

Follow-Up Hypothesis: News sources will generally cover topics relating to relief and ramifications of the fire.

We made this hypothesis because we believed that these two topics would be the two main aspects any natural disaster would report. All in all, we thought it would be logical for news sources to focus on efforts to help the ramifications of the fire and the ramifications itself.

Project Significance

This project is important because it can highlight how the information you receive can depend on the type of news source you are digesting news from. More specific to the Camp Fire, this project will also help understand what topics news mediums are reporting regarding the Camp Fire.

Web Scraping and API

Local News: Paradise Post

Before web scraping the Paradise Post for news regarding the Camp Fire, we used the sub category titled "Camp Fire" to receive local news primarily focused on the Camp Fire. With this link we planned on creating a function that took in this url and returned all relevant article links; however, the individual page was restricted to 10 articles unless "load more" was clicked at the page's bottom. We found that in the HTML, each "load more" added "/page/(number)" embedded within the link. With this in mind, we created a loop within the function, mult_pages(), to receive all the links, which inputted url and number of "load more" clicks, and outputted all articles. To extract the relevant information from each article, we created a function, single_article(), which took in specific article url and outputted url, text, title, and dates within a dictionary. Combining these two functions, we looped these two functions together to extract all relevant articles, using XPath, for the stated parameters. The number of "load more" clicks taken was determined to be 8, which led back to the beginning of February. This date restriction was taken due to the limitations of the Twitter API only being able to return a limited number of tweets.

National News: Google News

For national news, we scraped individual articles by using functions from Newspaper3K. The Newspaper3K package provided us a simple way to scrape the articles without having to write web scraping code for each news website. To retrieve the article links, we found the Google News links for nine relevant search terms. Some of these search terms included "camp fire california", "camp fire death toll", "camp fire investigation", etc. Each of these links direct to a web page that lists articles from the initial query. We created a function with xml paths to get the link of each article on the web page. The Newspaper3K package only needs the article links, so we compiled all the links into a list and created a function to scrape each articles. For each article, we obtained the title, text, and date, then built a dataframe from this information.

Twitter News

We also procured user Tweets using the Twitter Premium API. The API allowed us to query up to thirty days worth of tweets, and it returned 100 results per query. As a result, the time frame of the tweets was from the beginning of February to the start of March. To make use of the API, we used the Twitter-API Python package and provided it with our private user keys. The library included a function that allowed us to continuously gather tweets until we reached our limit of thirty days. Each query returned a JSON object of the fields we requested, and we stored each of these objects in a list of dictionaries. We queried for the text in the tweets and other potentially useful user data of the tweet such as location and username. Nonetheless, we were only able to use the text data because the other fields were often incomplete. In total, we obtained around 4800 tweets and converted the list of objects into a Pandas dataframe.

Data Cleaning and Storage

Paradise Post

When cleaning the 80 articles received by Paradise Post, we saw that much of the text had "\n" and "\t". To compensate for the new lines and tabs, we used a strip function for title and replace function for text to remove the white space.

Google News

Since we used multiple search terms to retrieve the article links, we had some instances of duplicate articles. To solve this issue, we subsetted the initial data frame so that all the article titles were unique. Furthermore, we also got articles from the Paradise Post in the Google News searches. We did not want overlap between our national and local news sources, hence, we removed articles from the Paradise Post as well.

Twitter

An issue we encountered with the Tweets is that many of the tweets contained undesirable elements such as HTTP links to other tweets and emoticons represented in Unicode. To ameliorate this issue, there needed to be a search using regular expression that made sure the token from the tweet did not contain special characters and contained text. Additionally, we removed duplicate tweets from the corpus.

Collective Methods

After individual methods, we saved each of the data frames created from local, national, and social media news sources, saved it as a pickle file, and combined them into one full data frame. Before cleaning, we first made sure to add unique "tweet ids" per tweet processed to make sure we were able to identify each individually. In addition, we made a new column called source, which categorized what news medium the article or tweet came from (categorized by either "local", "national", or "twitter"). After combining the data into a full data frame, we performed text cleaning by tokenization, standardization through lower-casing, lemmatization, and stop word removal of the text. For future loading in Pandas, we again used the package "pickle" to preserve the full data frame.

Exploratory Data Analysis

Word Cloud & Term Frequencies

For exploratory data analysis, we created word clouds and term frequency plots summarize what potential topics and content were in each news medium. For the term frequency plots, we used the already cleaned data to create a plot that illustrated the top 25 words prevalent in each news medium. From these plots we hypothesized that topic differences between each news medium are reasonable. We created a word cloud to better visualize this, which also included more than 25 words without compromising graph readability. All in all, we were able to compare these word clouds and frequency plots to understand that topics and frequency of those topics differed, which suggests that we could further analyze this concept. For example, Twitter included exclusive words regarding probable, cause, and community, national news had county, state, and destroy, and local news had property, town, and work. (reference Appendix)

Term Frequency-Inverse Document Frequency

After determining that there was some merit in investigating the differences in topics between the different news mediums, we created a term frequency-inverse document frequency matrix, or tf-idf matrix, to prepare us for further analysis and tell us how important a word is to a given document. We used scikit-learn's TfidfVectorizer to compute the tf-idf matrix. Once the tf-idf matrix was created, we used it to start the process of hierarchical clustering.

Hierarchical Clustering

Process

We used Ward's method to conduct hierarchical clustering analysis, a form of agglomerative clustering. Ward's method starts with single element clusters, then it merges the most similar (closest) pair of clusters into one cluster. This process continues until all elements are grouped into one large cluster. This is analogous to a tree; where the elements are the leaves, the sub-clusters are branches, and the big cluster is the trunk.

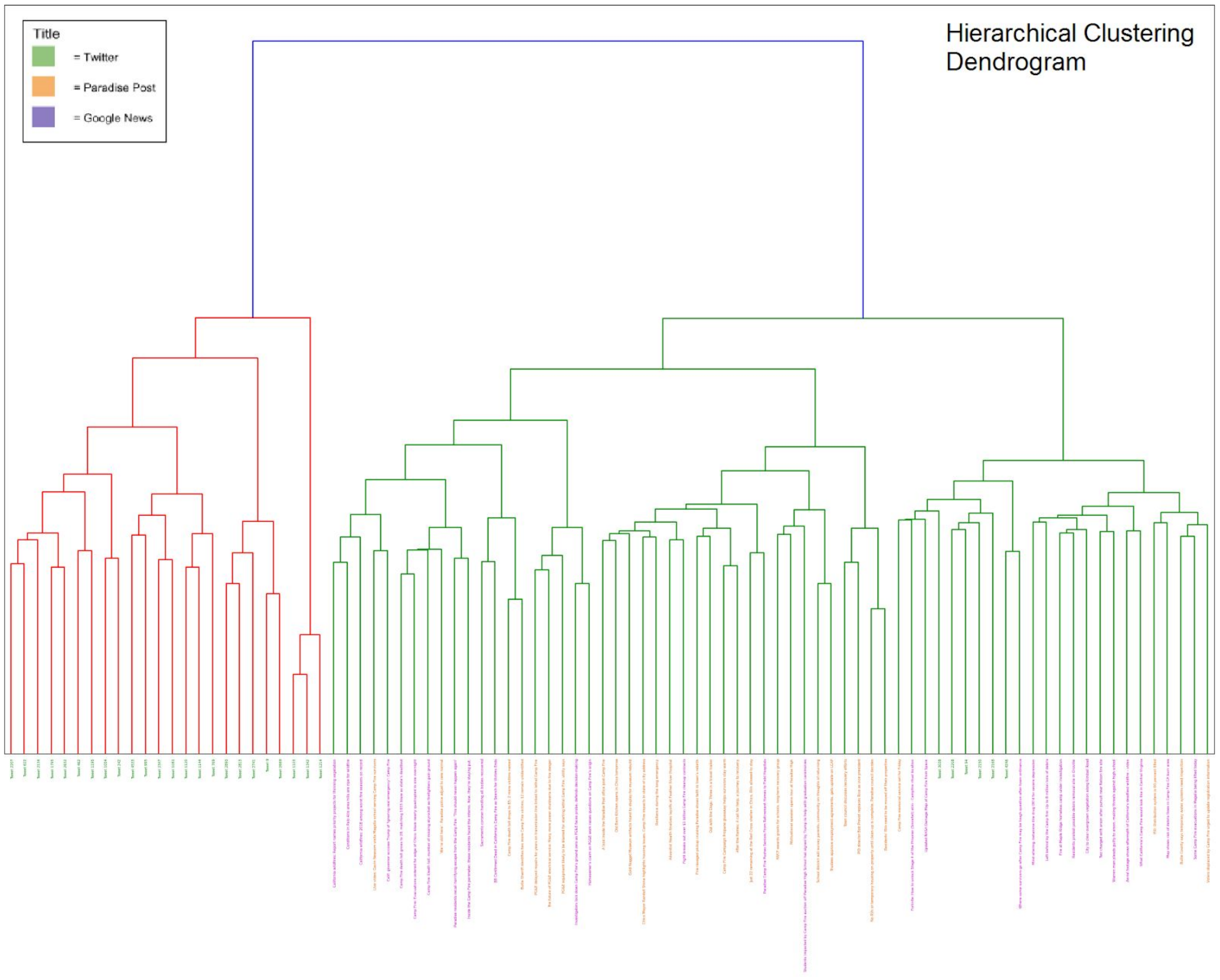
The parameters in the hierarchical clustering algorithm are the distance metric and linkage method. We used the cosine distances for the distance metric and Ward's method for the linkage method.
The similarity between elements is determined by the distance between elements. Ward's method defines this distance as the increase in sum of squares when two clusters are merged. In other words, the proximity between clusters is $SS_{12} - (SS_1 + SS_2)$, where SS_{12} is the sum of squares for the combined cluster, and S_1 and S_2 are the sum of squares for cluster 1 and cluster 2, respectively. As the cluster increases in size, the increase in sum of squares gets larger as well. Ward's method groups clusters in such a way that minimizes this increase in sum of squares.

To create the hierarchical clustering dendrogram, we took a random sample of 30 articles from each of the 3 news mediums to get a total of 90 observations and calculated the tf-idf matrix. The purpose of taking a sample was to ensure that the dendrogram would not be too large. Each article title or tweet name will be labeled on the dendrogram. Hence, if we used all of the observations, we would have a couple thousand labels on the plot.

Results

Hierarchical clustering allows us to see how clusters relate to other clusters. The dendrogram will show that some clusters are closely related, while other clusters are distantly related. The green labels represent tweets, the orange labels represent local news articles, and the purple labels represent national news articles.

From our hierarchical clustering dendrogram, we were able to see that there were two main clusters within the sample we took denoted by the green and red group of brackets. As shown, the green cluster is dominated by tweet articles and the red cluster is primarily dominated by a mix of local and national news articles. Since the local and national news articles are intermixed (not all the local articles are grouped together within the cluster and not all the national articles are grouped together within the cluster), this assumes that there aren't many differences between the local and national articles' contents being reported. All in all, this suggests that tweets have different content while national and local news tend to mirror each other. To further our analysis, we used K-means clustering to give us a better idea of what those clusters are. Before that, we re-plotted the dendrogram over different samples and found that the clusters varied from 2-4. Since the input of k-means requires specifying the number of clusters, we tested k-means with clusters from 2, 3, and 4 to start.



K-Means Clustering

Process

Along with hierarchical clustering, we wanted to group the documents using K-means clustering to see if we would get similar results. Another benefit of implementing the K-means clustering algorithm was to visualize the similarity of articles and tweets for the whole data set, instead of a small sample.

The K-means clustering algorithm is a form of unsupervised learning, meaning that the input data is unlabeled. The number of groups, denoted by K, is a parameter in the algorithm. K-means clustering assigns the data points into K groups, based on the similarity between data points. The similarity between documents was quantified by the tfidf.

We started with a corpus of text from local articles, national articles, and tweets. Recall that these articles and tweets do not have any labels that indicate which news source it came from. The articles and tweets were stored in a dataframe and we removed any duplicate articles or tweets. With this data frame, we were able to calculate the tf-idf.

To implement K-means, we used the K-means clustering function from scikit-learn. The input for this function is the tf-idf and the number of groups, K. The K-means function returns a number for each document; and this indicates which cluster the document belongs to. After running the K-means algorithm, we created a dataframe with the document and its assigned cluster.

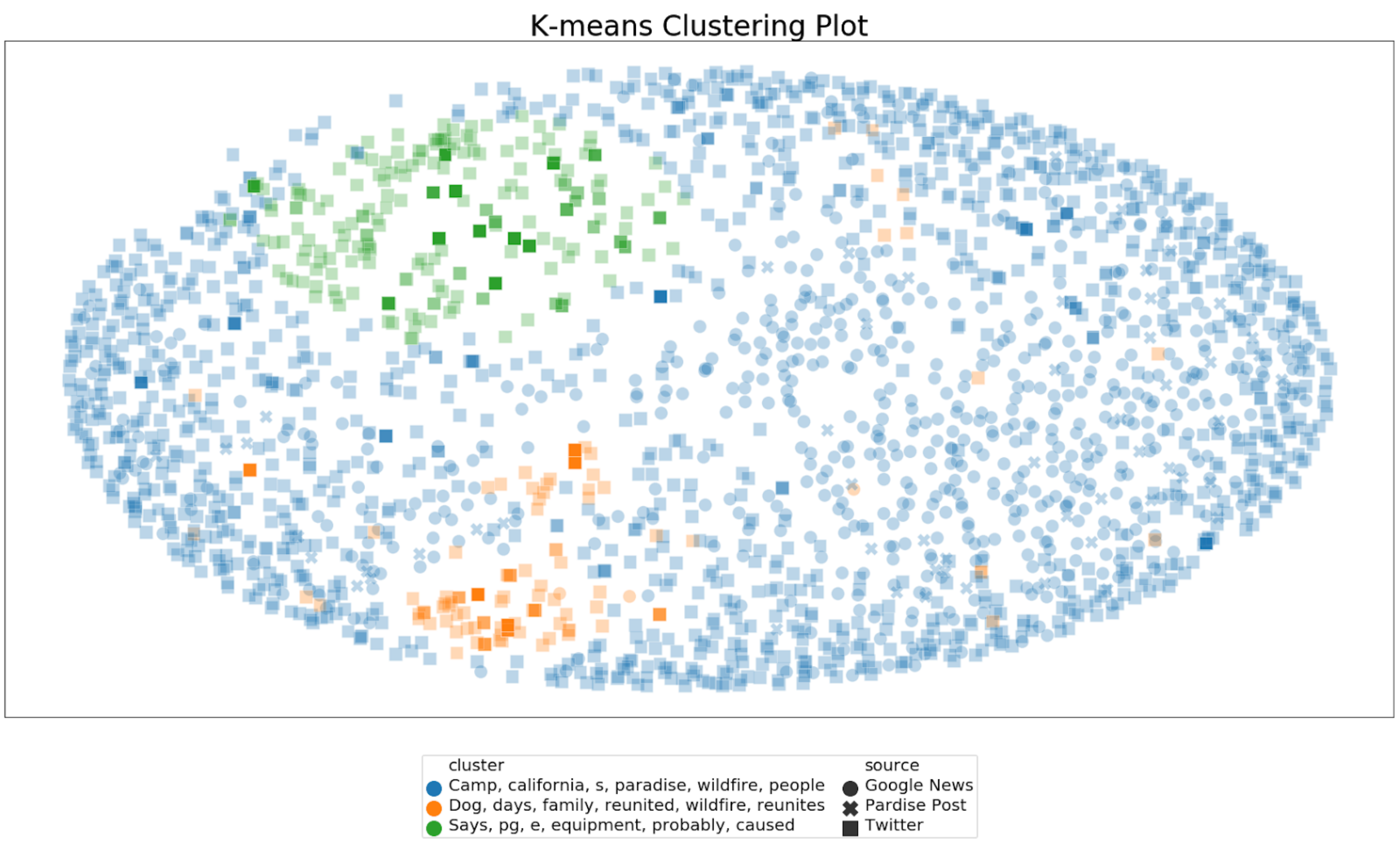
Now that we had the cluster assignments, we had to figure out a way to visualize the clusters. To do this, we used Multidimensional Scaling (MDS) to convert a distance matrix to a coordinate matrix. The goal of MDS is to visually represent a set of distances between documents in a lower dimensional space. With the tf-idf matrix and cosine similarity, we were able to calculate the cosine distance. Then we used MDS to transform the cosine distances and plot the points on a graph. Additionally, he labels for each cluster are the six words that are closest to the cluster centroid.

Results

From observing the hierarchical clustering dendrogram, we initially started with K = 4. In other words, the algorithm will group the documents into four clusters. In the plot with K = 4, there were two clusters that overlapped. Therefore, we tried K = 3 and the clusters were much more distinct in this plot. We also compared the plots for K = 3 and K = 2. The clusters seemed to be more general in the K = 2 plot, hence, we decided that K = 3 yielded the best plot.

When using 3 clusters, we were able to see that 2 clusters, the green and orange, were dominated by Tweets. Since the orange cluster is dominated by Tweets, this suggests the Twitter news source regarding the fire contains articles primarily discussing more personable news such as potentially lost dogs or family being reunited. As for the green cluster, Twitter news most likely contains a lot of discussion on PG&E since the news is not as credible and more opinion based. To be specific, the cause of the fire is still unknown; however, given the tragic 2008 San Bruno pipeline explosion, many people are speculating PG&E to be the cause of another significant disaster.

Furthermore, the plot shows that articles from local and national news sources overlap in the plot. Hence, the tf-idf values for these articles are more similar to each other when compared to tweets. This may imply that the content of local and national news articles parallel each other.



Latent Dirichlet Allocation

Process

Although clustering methods such as Hierarchical Clustering and K-Means aided in understanding how the mediums differ structurally, we needed to use a topic modelling technique called Latent Dirichlet Allocation (LDA) to determine the topic composition of the entire corpus. Before performing any topic modelling, we first needed to decide on three parameters: the number of topics, alpha, and beta.

One may select the parameters arbitrarily, but there exist methods to obtain a model that better reflects the topic and word distributions of the corpus. For the number of topics, we had the option of using the R package called “ldatuning.” The method called “FindTopicsNumber” performs some metrics over a range of number of topics and reports back corresponding values of the metrics. We created a plot of the values of each metric over the number of topics and examined which number of topics was the best fit. The “best” number of topics would be the number that maximizes our maximization metrics and minimizes our minimization metrics; we found this number to be around 40 topics.

For our purposes, we assumed a symmetric distribution. In this case, the parameter, alpha, represents topic-document density, and the parameter, beta, represents the topic-word density. With a higher alpha, documents contain more topics, and with lower alpha, documents have fewer topics. With a high beta, topics are consist of most of the words in the corpus, and with a low beta, few words in the corpus are in the topics. The range of these parameters is between 0 and 1. We assumed a low alpha since we believed that the there would not be too many topics related to the Camp Fire in the corpus. Additionally, we assumed a significantly lower beta value because we believed that there would not be a substantial amount of words that contributed to the topics. The alpha value was 0.08 and the beta value was 0.01.

To generate the model, we used a Java application called MALLET, which can perform topic modelling via the Unix command line. Firstly, we had to make a text file where each line is the cleaned and tokenized text of each document and import this file into MALLET. We decided to perform the topic modelling step in R, since R had a package called "LDAvis", which makes interactive visualizations on LDA topic models. Using the package RJava, we created a script that made Java calls to MALLET to perform the topic modelling, providing it with the parameters we previously selected. The output was an HTML page that directed to the visualization and a JSON file containing data to make the LDAvis.

Results

Note: To view the visualization, clone the repository → open *LDAVis/index.html*

Each point represents a topic, and one can observe the top-30 most relevant terms associated with that topic by hovering over the point. Compared to the results from the clustering methods, we can observe topics that include terms related to the aftermath of the fire such as "destruction" and "burn." In addition to these terms, there exist terms within these topics which suggest personable discourse such as "home" and "resident." Most of the clusters are within close proximity, which suggests that the relevant terms for the topics are closely related. Consequently, this may suggest that the documents in our corpus are quite homogeneous, in which they tend to discuss similar topics. While there was significant discourse for the consequences, there appeared to be no discussion for any relief or aid. As a result, our second hypothesis was slightly inaccurate in that we expected topics to both encompass destruction and relief, but the topic of destruction was the most apparent. Nonetheless, there was discussion of terms such as "debris" and "work", under the same topic (e.g. topic 16 and topic 39), which may suggest some topic of recovery after the fire. Overall, topics are quite similar to each other and tend to mention the aftermath of the fire.

Overall Summary

Hypothesis: Local news will approach coverage on the Camp Fire with a more episodic perspective compared to national and social media news sources.

Actual: Twitter news was the news source that actually reported news with a more personable or individual perspective relative to local and national news. In addition, local and national news seemed to mirror each other about topics and content being discussed.

As shown, our initial hypothesis was inaccurate. In hindsight, this actual observation does make sense since rather than coming from reputable news sources who try to report with credibility, twitter news is powered by the people who will report on topics like lost dogs and PG&E.

Follow-Up Hypothesis: News sources will generally cover topics relating to relief and ramifications of the fire.

Actual: All three of the news mediums actually reported mostly on ramifications; however, any topics relating to relief and help were barely mention across all three boards.

Our initial hypothesis was again inaccurate; however, partially correct. As mentioned within LDA's results, news sources did report on ramifications but LDA showed little mention of any topics relating to aid and recovery.

Limitations

When discussing the project in general, it is good to note that there were approximately 4800 tweets, 650 national articles, and 80 local articles. Keeping this in mind, certain news sources were more represented by others and for the K-means graph, which takes all observations, Twitter news is represented at a much higher rate than the other three news mediums.

In addition, we were limited to restraining the date of the articles from the beginning of February to current time due to the Twitter API only being able to extract Tweets from up to a month prior to the date of the query. Therefore, the scope of our project is limited to a few months after the California Camp Fire had been resolved.

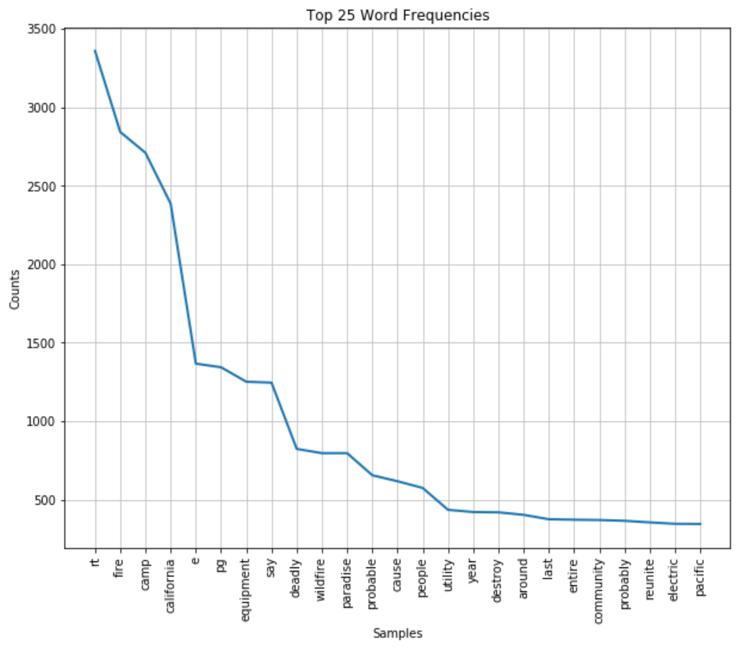
Next, when we created the topic model using Latent Dirichlet Allocation, we needed to manually select our parameters. While we had a rather empirical method of selecting the number of topics through the "ldatuning" package, the same options were not available to us for selecting alpha and beta parameters. We selected low alpha and beta parameters based on our assumptions about the topic and word composition of our corpus, but we did not have the opportunity to try out more values for these parameters.

Lastly, we believe Twitter news encompasses more topics, but this doesn't necessarily suggest that Twitter is the best place for one to receive news. Our project does not address the accuracy of each news source, and user-curated news from Twitter could be less reliable than actual news sources.

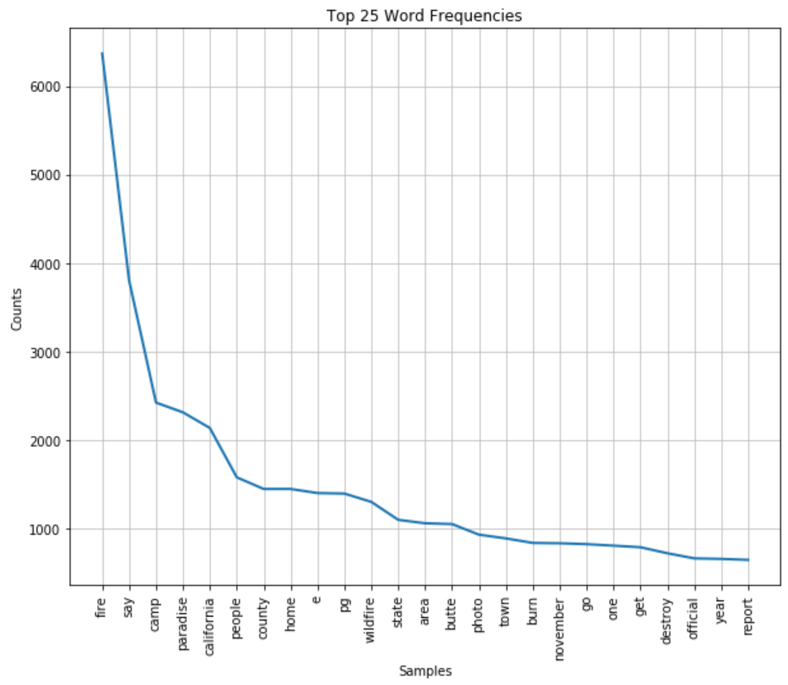
Appendix

- http://brandonrose.org/clustering
- https://www.thoughtvector.io/blog/lda-alpha-and-beta-parameters-the-intuition/
- https://opendatascience.com/using-the-newspaper-library-to-scrape-news-articles/
- https://www.geeksforgeeks.org/newspaper-article-scraping-curation-python/
- https://methodi.ca/recipes/document-clustering-topic-using-k-means-and-mds
- https://rstudio-pubs-static.s3.amazonaws.com/93706_e3f683a8d77244a5b993b20ad6278f4b.html
- https://www.stat.cmu.edu/~cshalizi/350/lectures/08/lecture-08.pdf
- https://newonlinecourses.science.psu.edu/stat505/node/146/

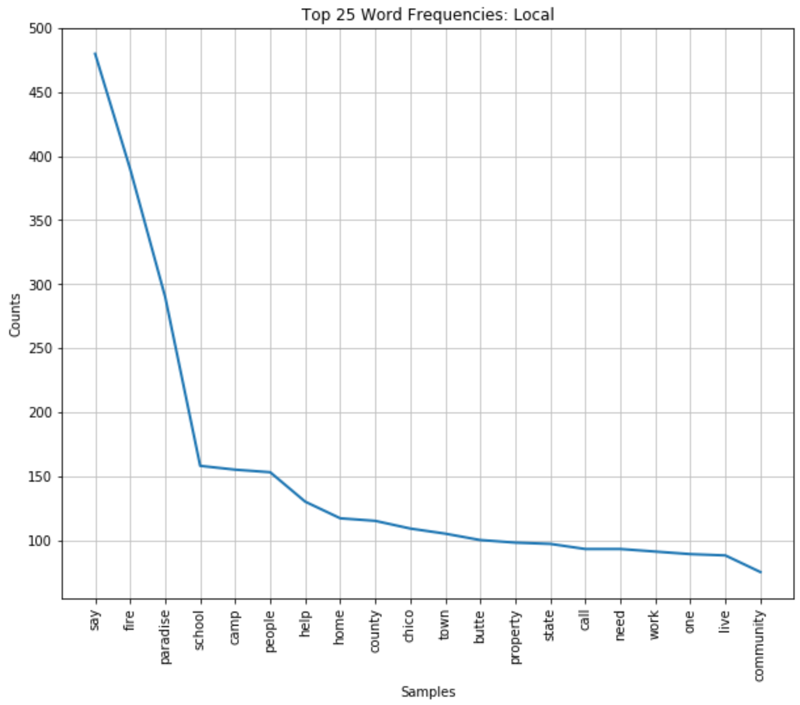
Twitter: Frequency Plot



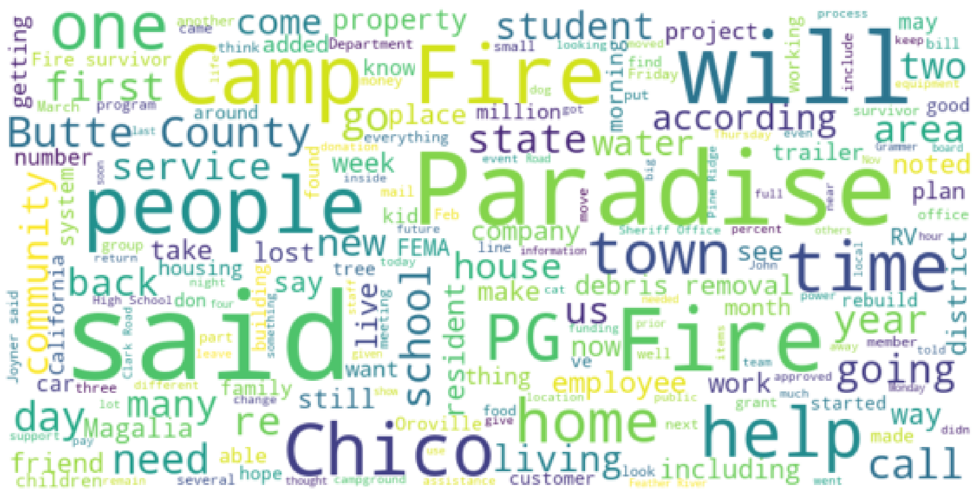
Google News: Frequency Plot



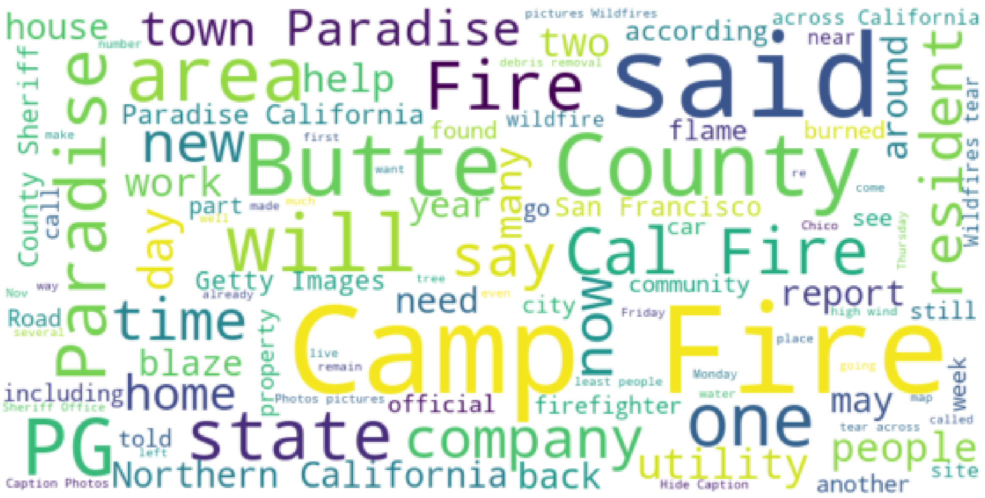
Paradise Post: Frequency Plot



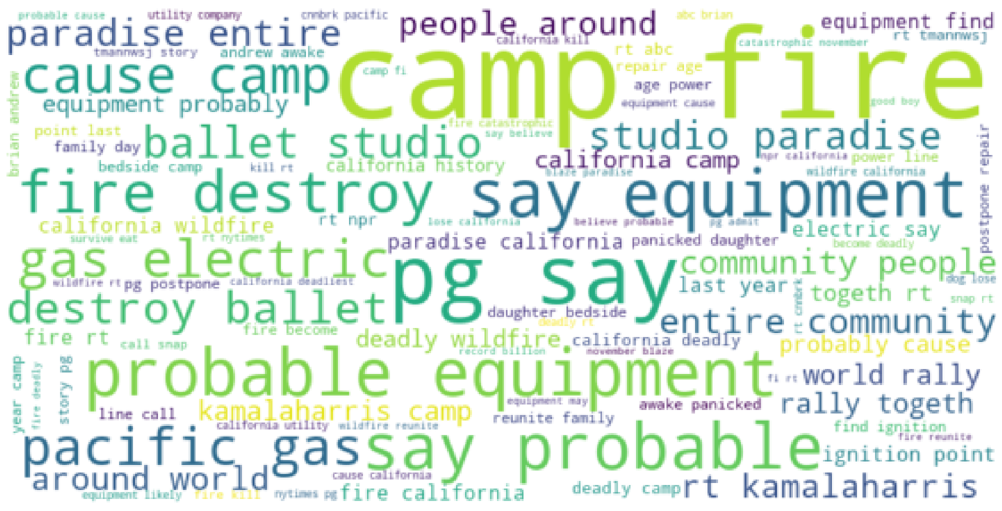
Paradise Post: Word Cloud



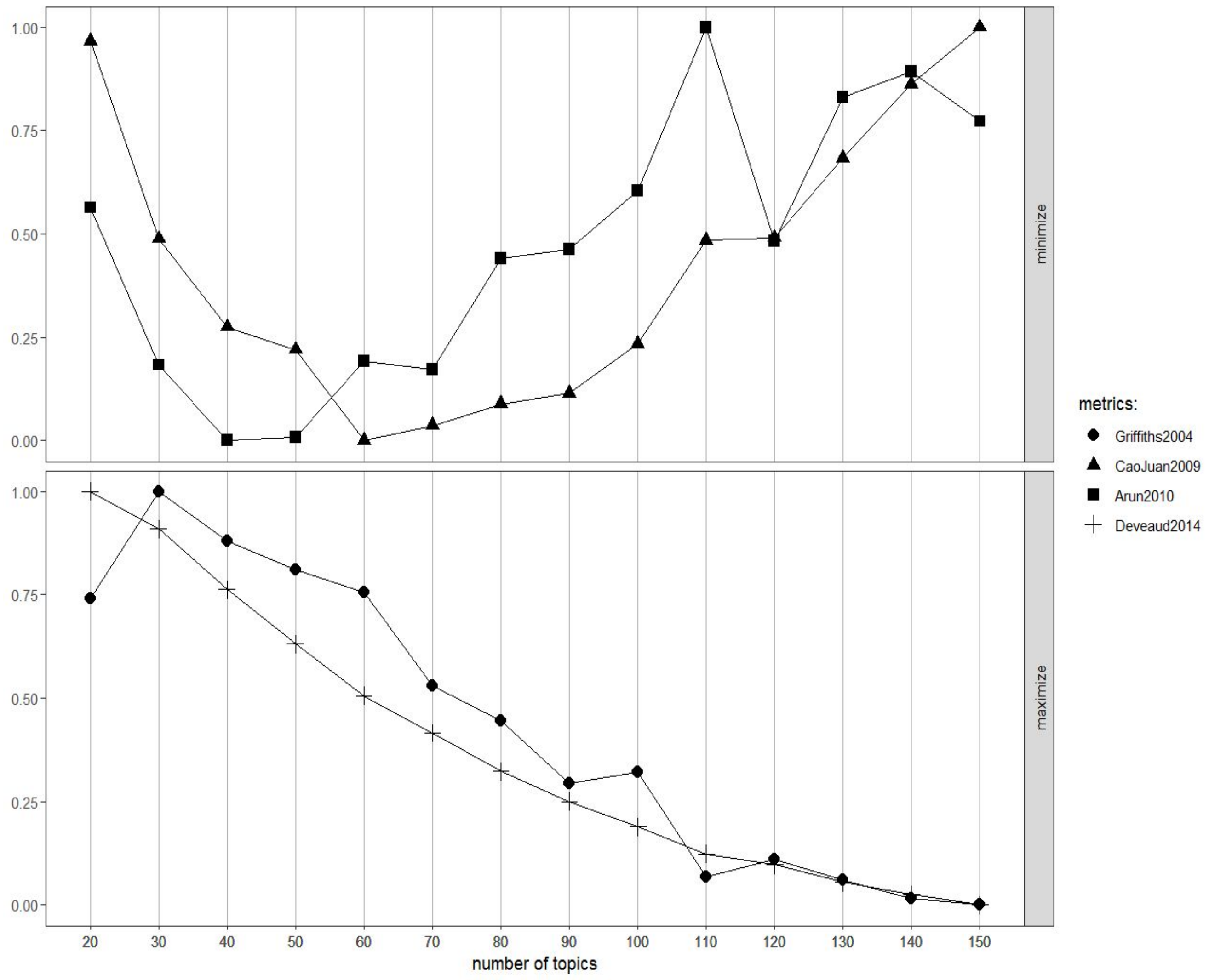
Google News: Word Cloud



Twitter: Word Cloud



LDATuning Plot


minimize
maximize
number of topics
metrics:
● Griffiths2004
▲ CaoJuan2009
■ Arun2010
+ Deveaud2014