Tweet relevance analysis on the basis of hashtag

Shivani Matta
2018191
IIITD
shivani18191@iiitd.ac.in

Mukul K. Rajak
2018054
IIITD
mukul18054@iiitd.ac.in

Tejas Dubhir
2018110
IIITD
tejas18110@iiitd.ac.in

Anmol Kumar
2018382
IIITD
anmol18382@iiitd.ac.in

Rakshit Singh
2018079
IIITD
rakshit18079@iiitd.ac.in

ABSTRACT

In this project, we tried to classify legitimate and hijacked tweets/trends by analysing and evaluating text corpus and common spam account attributes under popular hashtags using NLP and machine learning techniques. We curated our own dataset for tweets using Twitter Search and Streaming API for both historical and real-time data. We further introduced a new attribute of scoring popular user sentiment and evaluating all the tweets under a particular hashtag/campaign to improve upon existing methods. We use tfidf-vectorizer, Count-vectorizer and tweet information(User's friends, followers, retweet count, and favorites count) to train the ML models, and from their output we vote out the best results. We train the following ML models with the above mentioned data: K - nearest neighbour, Support Vector Machine, logistic regression, XG boosting classifier, Gradient boosting, Ada-boost classifier, Random forest classifier, Extra Tree Classifier, Neural Network, Gaussian naive bayes, Multinomial naive bayes classfier.

1. INTRODUCTION

Twitter is one of the most active and open platforms for sharing information and opinions on any subject matter. Since the onset of hashtags, online communication has become trend-driven. It has also helped in making the content on the platform somewhat structured. People can choose to follow any trend popular at any location in the world. Hashtags drive people's attention and affect their opinions, help them find people with similar interests, or even build an audience for marketing. Brands, political parties, and other organizations try to leverage hashtag promotions for campaigns and endorsements. But often relevant and meaningful twitter trends are targeted through spam and other content under the same hashtag pushed by a legitimate user. This is termed as Hashtag Hijacking. Spammers, tweet farms and bots try to bypass twitter restrictions and guidelines making people more exposed to privacy and security issues, fake news, etc. and users are always at a risk of being targeted for trolling and cyberbullying.

Hashtag hijacking has been successfully implemented by some brands for reaching a wider audience or getting more views. But most of the times, well, the marketing team suffers from a miserable ”fail” Brand example. A difficult problem for Twitter is to flag hijacked tweets from accounts with lesser interaction Spread of malicious and spam content through links is easily possible because external web documents are outside the scope of the Twitter ecosystem to process. Many sensitive topics and important topics like mental health get diverted. JusticeforSSR and FarmersProtest were hijacked for personal agenda by certain people/parties. This discourse has promoted the spread of fake news/manipulated media, bullying, trolling, and slut-shaming among many other problems. Some trending hashtags are just used in the tweets to get better visibility for promoting irrelevant events.

1. Alllivesmatter : During the protests in the United States in Sep 2020, the hashtags BlackLivesMatter and AllLivesMatter were trending on twitter, and instantly due to their fame, K-Pop fans hijacked the hashtag AllLiveMatter to defuse it and portray it in a whole new meaning.

2. McDStories : This is the most famous example of hashtag hijacking, where McDonalds made the hashtag McDStories for people to share their memories made with McDonalds, but people hijacked the hashtag to portray the fast food chain in an unhealthy way.

2. LITERATURE REVIEW

HashMiner does feature characterisation and analysis of hashtag hijacking using real-time neural network.[1] The paper characterizes the increases the cases of misuse of hashtags in the following ways:

1. Using trending hashtags to advertise some specific brand to gain popularity through the hashtag.

2. Users post unrelated content to distract the population from the main topic and distribute spam, fake and unwanted content.

3. Some of the hashtags are also hijacked to post political opinions which can sometimes be sensitive to the users.

For this paper, the authors have collected 10240 tweets and classified them for 10 hashtags in totality.

Jain, N. et al. presents a general algorithm to detect and analyze hijacked tweets from Twitter.[2] The paper proposes an algorithm that divides the collected datasets into various categories like entertainment, politics etc. This categorization is used to detect and analyze the hijacked tweets. The researchers collect tweets from Twitter and manual annotation of the tweets is done to prepare training and testing dataset. The dataset is used to find the top occurring words and assign tf-idf score to the top words in the dataset. These top scoring words are then saved into a dictionary corresponding to each category. A score is calculated for the new incoming tweets by matching the tokens in the tweet with words in these dictionaries. For each math a score of 1 is assigned. The algorithm outputs score for the tweets and high scoring tweets signify that the tweet is not hijacked whereas if a tweet has a low score then it is a misleading/hijacked tweet. The research paper however does not deal with URLs and is hence unable to detect hijacked tweets if they only contain a URL and no text.

VanDam, C. and Tan, P. 2016 have proposed a novel framework that combines information about the temporal distribution of hashtag frequencies and the content of their tweets and the users who posted the tweets to determine whether a hashtag has been hijacked.[3] They have selected around 2667 trending hashtags that have appeared in at least 100 tweets on the same day using Tweepy from 766,057 unique users and applied NLTK to 98,234 unique nouns from the data. They have generated 2 data matrix for each hashtag (i.e., X:a term frequency per day and U: a user frequency per day.) and normalized by frequency of tweets, which further decomposed into 3 latest factors (i.e., W: a terms by topic matrix, V: a user by topic matrix, H: a day by topic matrix) and detected hashtag hijack by employing a multimodal non-negative matrix factorization approach to learn each hashtag's underlying topics by optimizing the following function:

$$\|XWH^T\|_F^2 + \alpha\|UVH^T\|_F^2$$

s.t. $\mathrm{W}_{ij} \geq 0, V_{kj} \geq 0, H_{hj} \geq 0 \ \forall \ h, i, j, k$

They have also applied an alternating minimization approach to estimate the latest factors W, V, H. And then they have updated the matrix value until they converge using:

$$W_{ij} = W_{ij}(XH)_{ij}/(WH^TH)_{ij}, V_{kj} = V_{kj}(UH)_{kj}/(VH^TH)_{kj}$$

$$H_{hj} = H_{hj}(X^TW + \alpha U^TV)_{hj}/(HW^TW + \alpha HV^TV)_{hj}$$

This was followed by Hotelling's t2 (a popular change detection algorithm) test to look for the hijacked topic. The hashtag that fails this test has been considered hijacked. They have found that a window size of 3 days gave the best accuracy of 66.9% when user matrices are decomposed into 5 topics.

Spam detection and hashtag hijacking have many common research goals by principle. Chu, Z. et al. 2012. focuses on detecting spam-driven campaigns rather than focusing on individual tweets and accounts.[4] Their approach is more effective and robust as analyzing individual tweets is computationally more resourceful while hardly meeting stringent requirements. Also, spammers usually tend to make multiple accounts to stay under the radar to avoid detection, but their end goal is to get maximum reach and engagement in the form of content, which is hashtag or trend-driven in nature. In this study, the researchers collected 50 million tweets posted by 22 million users with the help of Twitter API and partitioned the tweets containing URLs based on campaigns. Spamming URLs with unrelated hashtags are one of the most common methods for the purpose. URLs are usually shortened by online tools to avoid spam blacklists with an added layer of multiple URL hops. For this, the researchers designed an extension that extracted the last hop URLs and compared them for affiliate/referral dissemination. Scripted automation along with human inspection for spam content / unrelated URL content/duplicate or similar content via multiple or single accounts was used for annotating the dataset. Further, many custom features based on Twitter parameters were designed, with the help of techniques like tf-idf scoring and 2nd order similarity scoring (to detect synonym interchanging). Ultimately, a Random forest with cross-validation scoring was chosen as the machine learning algorithm for its ensemble characteristics. Although the FPR and FNR are comparable, the method is highly efficient for real-time detection, considering the amount of tweets made per unit time on the platform. The main advantage of this approach is that it is trend or hashtag driven. But the major limitation is that only URL have been primarily been the focus of this study. Other parameters related to sentiments could have improved the performance metrics without increasing much of the complexity.

Hadgu, A. et al. 2013. present present the study on the polarization of hashtags on Twitter and then shows how certain jumps in polarity are caused by hijackers.[5] They have used retweets of labelled seed users (mainly political leaders) to obtain political orientation of users. By analyzing their hashtag usage, leaning to hashtags is assigned. “Change points” with a sudden jump in leaning are identified which is due to the activity of “hashtag hijackers”. Their methodology first identifies seed users and then the tweets scanned for hashtags. Retweeting people are identified and filtered based on their location. After that, leaning with respect to a party is defined using lean formula and validated against wefollow.com, twellow.com and persecuting data. At last, outliers are marked as change points from which hijacker points are identified. The shortcomings of this paper are that URLs, images and sarcasm cannot be detected by their model and thus, we will have to find a way to include these too.

3. METHODOLOGY

We used the Twitter API to collect the tweets corresponding to the hashtags which are trending. For that, we had to apply for a Twitter's developer account which usually takes weeks to be approved. We were asked about the project and about how we were going to use the collected data, once we replied to them about the details of our work, we got access to the public tweets. The tweets obtained had a lot of information along with the text, such as username of uploader, his/her followers and friends count, tweet time of uploading, retweet information, location of tweet if available and many more. These tweets were used to create the dataset. Another challenge we faced was that there could only be 100 tweets extracted through the API. The tweets were manually annotated into categories of whether the tweet is hijacked or not. Our initial goal was to collect 10,000 tweets and annotate them to create out dataset. Since tweets can be subjective and it depends on the person whether he/she considers it to be hijacked or not, we followed a majority voting approach. Each of the tweet has been manually read and classified by each and every team member. Out of 5, if more than 2 members need to agree to classify a tweet hijacked. This way we removed opinionated bias. Moreover, there were numerous hashtags and their respective popularity's were very dynamic. Ultimately we were able to create a dataset of 9470 samples.

The tweets were collected using Twitter API as mentioned above and the tweets were retrieved as JSON objects and stored as .csv file. As some tweets had a few columns values as NaN or missing values, when these tweets were stored into a csv file, their column orders were skewed. For example: the User column entry actually got shifted to Geo coordinates columns. This had to be resolved by running a script to iterate over all the columns of the tweet to find the corresponding shifted values. Moreover, since the tweets were collected in batches over a duration of 4-5 days because of the rate limit of twitter API, their format differed a bit. So the script for filtering out the relevant attributes for each batch of tweets varied a little. In the final dataset, we had just the columns named label, full-text, created_at, screen_name, followers_count, friends_count, retweets_count, favorites_count. The rest of the features were not relevant to our goal.

After the creation of the dataset of sufficient size, we proceed with dataset cleaning and using NLP techniques like tokenization, stop-words removal, and stemming/ lemmatization. Techniques like tf-idf, ranking algorithms, and vector space models are used for scoring words. In addition, we analyzed if the sentiment of the current tweet matches with the sentiment of the majority of tweets of that hashtag. Then the weighted cumulative score from both these parameters helped us in voting the final outcome to classify the nature of the tweet.

We then applied various techniques on the obtained posts or tweets and obtain the sentiment, retweets/reposts, likes, uploader, uploader's data (like followers, following, verified or not), related hashtags (their relative scores), and the tf-idf score of the post. The Machine Learning models we used are K - nearest neighbour, Support Vector Machine, logistic regression, XG boosting classifier, Gradient boosting, Ada-boost classifier, Random forest classifier, Extra Tree Classifier, Neural Network, Gaussian naive bayes, Multinomial naive bayes classifier. We let all the ML models run through the dataset we prepared, which contains a detailed examination of the posts and annotated classes (hijacked or not). After which we used the obtained weights to validate the rest of the dataset which will ultimately return us an accuracy, recall, F1- score, precision measure, and their macro-average and weighted average.

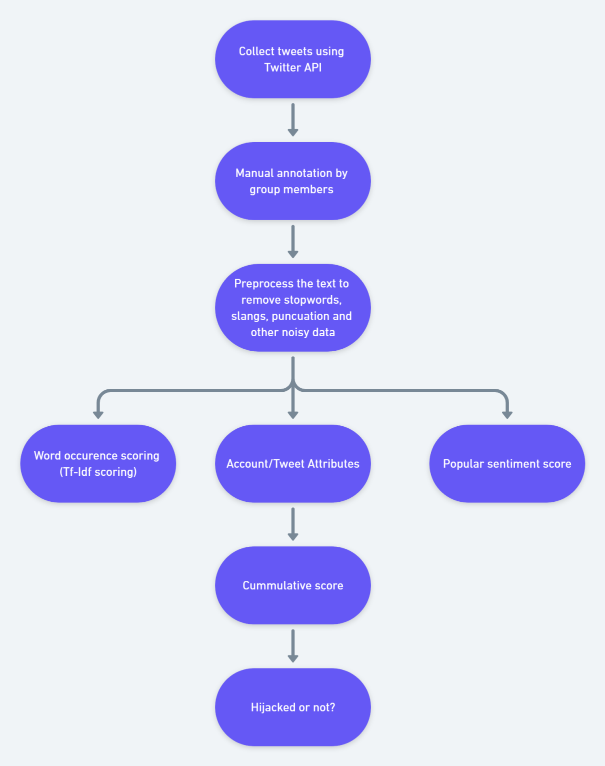
Collect tweets using Twitter API
Manual annotation by group members
Preprocess the text to remove stopwords, slangs, puncuation and other noisy data
Word occurence scoring (Tf-Idf scoring)
Account/Tweet Attributes
Popular sentiment score
Cummulative score
Hijacked or not?

Figure 1: Procedure of our project

4. EVALUATION

As the tweets were manually annotated, we observed that the hijacked tweets were usually requesting for retweets/ likes /comments in a given time frame, or trying to obtain more subscribers/followers on some given account, also known as spam. Some tweets are trying to advertise their products. They used the popularity of the trending hashtag as a way to get their motive fulfilled and waste the space and time, of the reader if there was a hashtag related relevant post instead.

For baseline model, we trained a multinomial naive bayes model on the preprocessed full-text of lemmatized words. Then we had trained our model on dataset which contained 1072 manually annotated tweets. We achieved the accuracy of 58.4%. Which was improved when we finally ran this(and many more) model(s) on the complete dataset after analysing the user's data along with the tweet information. After the whole dataset was created and the models were trained, we obtained an accuracy of 95.0% with multinomial naive bayes.

Various techniques have been applied on the given posts/ tweets and obtain the sentiment, retweets/reposts, likes and dislikes, uploader, uploader's data(like followers, following, verified or not), related hashtags. We then let a neural network along with different models like random forest, logistic, SVM, KNN, multinomial Naive Bayes, and 3 boosting classifiers like XGB, Adaboost and gradient boost run through the combined dataset which we have prepared manually and

Table 1: Evaluation results by Tweet and user information

Model Name	Accuracy	Precision	Recall	F1-Score	Support
K-Nearest Neighbours	0.88	0.90	0.89	0.89	4472
Support Vector Machine	0.63	0.41	0.64	0.50	4472
Logistic Regression	0.36	0.13	0.36	0.19	4472
XG boosting classifier	0.976	0.98	0.98	0.98	4472
Gradient Boosting	0.978	0.98	0.98	0.98	4472
Ada-boost Classifier	0.78	0.81	0.79	0.77	4472
Random Forest	0.979	0.98	0.98	0.98	4472
Extra-tree classifier	0.981	0.98	0.98	0.98	4472
Neural Network	0.66	0.70	0.67	0.58	4472
Gaussian Naive Bayes	0.36	0.77	0.36	0.19	4472
Multinomial Naive Bayes	0.43	0.68	0.43	0.35	4472

Table 2: Evaluation results of Count vectoriser

Model Name	Accuracy	Precision	Recall	F1-Score	Support
K-Nearest Neighbours	0.927	0.93	0.93	0.93	4472
Support Vector Machine	0.9787	0.98	0.98	0.98	4472
Logistic Regression	0.978	0.98	0.98	0.98	4472
XG boosting classifier	0.976	0.98	0.98	0.98	4472
Gradient Boosting	0.979	0.98	0.98	0.98	4472
Ada-boost Classifier	0.874	0.88	0.87	0.87	4472
Random Forest	0.923	0.93	0.92	0.92	4472
Extra-tree classifier	0.911	0.92	0.91	0.91	4472
Neural Network	0.978	0.98	0.98	0.98	4472
Gaussian Naive Bayes	0.86	0..90	0.87	0.87	4472
Multinomial Naive Bayes	0.95	0.96	0.96	0.96	4472

Table 3: Evaluation results of tfidf vectoriser

Model Name	Accuracy	Precision	Recall	F1-Score	Support
K-Nearest Neighbours	0.949	0.95	0.95	0.95	4472
Support Vector Machine	0.98	0.98	0.98	0.98	4472
Logistic Regression	0.956	0.96	0.96	0.96	4472
XG boosting classifier	0.977	0.98	0.98	0.98	4472
Gradient Boosting	0.976	0.98	0.98	0.98	4472
Ada-boost Classifier	0.881	0.89	0.88	0.88	4472
Random Forest	0.912	0.92	0.91	0.91	4472
Extra-tree classifier	0.892	0.90	0.89	0.89	4472
Neural Network	0.977	0.98	0.98	0.98	4472
Gaussian Naive Bayes	0.869	0.90	0.87	0.87	4472
Multinomial Naive Bayes	0.955	0.96	0.96	0.96	4472

up-sampled it to remove class imbalance, which contains a detailed examination of the posts and annotated classes(hijacked or not). The obtained weights were used to validate the rest of the dataset which ultimately returned us an accuracy and precision measure. We then used all these model and applied voting classifiers on these classifiers to obtain the best generalised model to avoid overfitting and saved this model and other individual best models using pickle for future use. After combining the results obtained from the models we obtain an accuracy of 0.981 through SVC.

After calculating the cumulative score of all the above 30 models by the voting classifier, we obtain the following statistics:
Accuracy: 0.981
Precision: 0.98
Recall : 0.98
F1-score: 0.98
Support : 4472

After sentiment analysis on the Full_text of the complete dataset after classification, we get:
Positive sentiment in Hijacked tweets 49.5098039215 %
Negative sentiment in Hijacked tweets 5.14705882352 %
Neutral sentiment in Hijacked tweets 45.343137254 %
Positive sentiment in UnHijacked tweets 33.479718016 %
Negative sentiment in UnHijacked tweets 17.31191494 %
Neutral sentiment in UnHijacked tweets 49.208367040 %

In terms of results obtained by giving new data, if a tweet has promotional/advertisement intentions, then the final may predict its relevance value. But, if the tweet has the content which is dependent upon the context to be classified as relevant or not, there needs to be some previously mentioned samples in the training data which make it clear for the model about the nature of the hashtag which has to be classified as relevant/ irrelevant.

5. CONCLUSIONS

We trained K - nearest neighbour, Support Vector Machine, logistic regression, XG boosting classifier, Gradient boosting, Ada-boost classifier, Random forest classifier, Extra Tree Classifier, Neural Network, Gaussian naive bayes, and Multinomial naive bayes classifier for the Twitter user's and tweet's data, Text of the tweet by transforming it to the word count vectors and tf-idf vectors. After running those models on the testing set, the best outcome obtained was an accuracy of 0.98, using SVM classifier which was trained using the tf-idf vectors of the text of tweets. After combining the results obtained from the models we obtain an accuracy of 0.981 which means the post voting results are better and are used instead of any single method. All this was done on a dataset collected, annotated and compiled manually by the group members.

6. REFERENCES

[1] Virmani, D. et al. 2017. *HashMiner: Feature Characterisation and analysis of Hashtag Hijacking using real-time neural network.* Procedia Computer Science. 115, (2017), 786-793.

[2] Jain, N. et al. 2015. *HashJacker- Detection and Analysis of Hashtag Hijacking on Twitter.* International Journal of Computer Applications. 114, 19 (2015), 17-20.

[3] VanDam, C. and Tan, P. 2016. *Detecting hashtag hijacking from Twitter.* Proceedings of the 8th ACM Conference on Web Science. (2016).

[4] Chu, Z. et al. 2012. *Detecting Social Spam Campaigns on Twitter.* Applied Cryptography and Network Security. (2012), 455-472.

[5] Hadgu, A. et al. 2013. *Political hashtag hijacking in the U.S.* Proceedings of the 22nd International Conference on World Wide Web - WWW '13 Companion. (2013).

[6] Rodak, O. 2019. Hashtag hijacking and crowdsourcing transparency: social media affordances and the governance of farm animal protection. Agriculture and Human Values. 37, 2 (2019), 281-294.