# G.R.I.D - Get Relevant Information on Disasters

Milestone Report

Deepti Aggarwal
Virginia Tech
Blacksburg, VA
deeptiag@vt.edu

Ananya Choudhury
Virginia Tech
Blacksburg, VA
ananya@vt.edu

## ABSTRACT

*In this document we propose a method to classify tweets-with-url related to disaster based on the 140 characters of tweet and after fetching data from those URLs, create a web archived search engine related to a specific disaster. The volume of the archived information on disasters is usually very high and filled with noise making it difficult to get information specific to a particular disaster. Hence we put forward a technique to successfully filter information related to disasters and index them thereby creating an archived search engine. We propose to classify tweets, expand tweet URLs, extract data from the URLs and index them in Solr. The resultant search engine, G.R.I.D will be useful for disaster management and post disaster analysis.*

## Categories and Subject Descriptors

[**DLRL Lab, Virginia Tech**]

## General Terms

Theory, Project Milestone

## Keywords

Disaster, Information Retrieval, Text Classification, Social Media Analysis, Microblogging

## 1. INTRODUCTION

Twitter is a fabulous source of information. It is a microblogging service that has become a new information channel for users to receive and exchange information. With brevity guaranteed by a 140-character-message limitation and the popularity of Twitter mobile applications, users tweet and retweet instantly. Everyday, nearly 170 million tweets are created and redistributed by millions of active users.

As tweets are created in real time, in recent years this social media platform has acted as an active communication channel in times of emergency as a result of which voluminous amount of data is generated. Processing such big data to obtain relevant information involves multiple challenges including handling information overload, filtering credible information and categorizing data into different classes. Another significant challenge twitter data analyst face today is the filtering of data between relevant and non-relevant as the number of spam tweets have increased exponentially with the increase of Twitter's popularity.

The goal of our work is to provide an efficient way to classify tweets into relevant and non relevant based on the tweet text and extract URLs from those tweets related to a disaster . We then fetch important text from the corresponding webpages and index it to Apache Solr for fast and efficient information retrieval.

Our approach is unique because our final goal is to create a web archived search engine specific to disasters. We also provide two step filtering: first classifying tweets and then extracting relevant texts from the web pages referred to in those tweets thereby providing a whole new dimension to the first step by using the tweet URLs as meta information sources for the tweet labeling.

## 2. COMMENTS FOR PROPOSAL

Below are the response to the comments made in the proposal.
- Detailed Project Management plan:

- Created training and test data: - 2200 tweets (by Deepti) (Time: 3 days)
  - 1800 tweets (By Ananya) (Time: 1 day)
- Feature Selection and Model Selection (completed by Deepti and Ananya)
- Classification Code Implementation :
  - 60 % implementation completed by Ananya
  - 40 % implementation completed by Deepti
- Expansion of URL in MapReduce (by Deepti) (Time : 3 days)
- Extraction of text from the expanded text (by Deepti) (Time: 9 days)
- Index data in SolR(by Ananya) (Time: 8 days)
- IBuild a simple UI(by Ananya) (Time: 4 days)

- Suitable references have been added as required.
- Details of the dataset have been provided under the section named dataset.

## 3. PROBLEM STATEMENT

In recent years, Twitter has been used to spread news about casualties and damages, donation efforts and alerts, including multimedia information such as videos and photos [8]. In November 2012, Twitter revealed that over 2 million tweets had been posted during Hurricane Sandy. In 2011, over 5,500 tweets were posted every second following the tsunami and earthquake in Japan (Anderson, 2012). Users posted over 2 million tweets about Haiti following the earthquake in January 2010 ‘ [6].In general, a significant amount of data gets generated during times of a disaster, some of which are extremely valuable for any post-disaster analysis.We provide an efficient way to retrieve and search information related to a crisis as the messages generated during the disaster vary greatly and a majority does not contribute to the disaster awareness.

**OUR APPROACH:**
We propose a 5-step method for disaster related information archiving:

- Step 1: Classification of twitter tweets into relevant and not-relevant categories using the Naive Bayes classifier.
- Step 2: Expansion of the URLs in the relevant tweets using Map Reduce
- Step 3: Extraction of text from the expanded links (webpages)
- Step 4: Index the data in Apache Solr.
- Step 5: Build a simple interface "G.R.I.D" for query search.

## 4. DATASET

We are using dataset which contains tweets related to Ebola in multiple languages such as English, Spanish, French, etc. This tweets are collected from March 29, 2014 available in http://cinnamon.dlib .vt.edu/twitter/. The entire dataset is received from Digital Libraries Research Lab, Virginia Tech.There are about 65550 tweets that we are currently working on.We created a sample of 4000 tweets by randomly collecting 4 sets of 1000 tweets across multiple timelines.

## 5. PREPROCESSING

The goal of the preprocessing part is to get the tweets in English language and which contains URLs.The URL string in the tweet should be of minimum 10 characters from the rightmost backslash to be considered for further processing.

## 6. DATA ACQUISITION

We will extract the required information for our data modeling. The dataset is a .csv file containing following information for each tweet: archive source, text, to_user_id, from_user, id, from_user_id, iso_language_code, source, profile_image_url, geo_type, geo_coo rdinates,_0, geo_coordinates_1, created_at, time. In order to get the required information we used the pandas package in python to read the file.We created a parser that identifies tweets with only valid URLs. We manually created the learning dataset for the classifiers by classifying the sample data of 4000 tweets into relevant and non relevant labels.

## 7. FEATURE SELECTION

Feature selection is an important component of classification. A flow chart of the various steps in classification process is added in Figure[1]. In this project, we classify tweets into two categories: relevant and non-relevant. We convert the collection of document into a matrix of TF-IDF features. Maximum features that can be extracted is 15000. From this set, we selected best 1500 tweets using SelectKBest method. The statistic used for feature selection is Chi-square.

Chi-square ($\chi 2$) test measures dependence between stochastic variables and does not consider features that are the most likely to be independent of the class and therefore irrelevant of classification. This statistic measure the difference between the observed counts $n_i$ and the expected counts $e_i$.

$$\chi^2 = \sum \frac{(n_i - e_i)^2}{e_i} = \sum_{i_1=1}^{m_1} \sum_{i_2=1}^{m_2} \cdots \sum_{i_d=1}^{m_d} \frac{\left(n_{i_1,i_2\ldots,i_d} - e_{i_1,i_2\ldots,i_d}\right)^2}{e_{i_1,i_2\ldots,i_d}} \quad (1)$$

We manually analyzed our sample 4000 tweets and identified 68 features which are relevant to classify text into relevant and non relevant. We added these 68 terms into the previous 1500 terms identified as features.

We considered retweets (tweets with RT tag in the beginning) as one of the features. We also considered the hashtags of relevant terms such as (# Ebola, #outbreak, #disaster, etc. ) or hashtag of mostly affected regions (# Africa, # Liberia, etc) or twitter account of corporate houses ( reuters , cnn, etc.) as few of the features. Unigrams, Bigrams and Trigrams are also extracted as features. All text in the corpus are converted into lowercase to avoid repetition of features in the TfIdfVectorizer method.

## 8. CROSS-VALIDATION

**K-fold Cross-validation:** K-fold cross validation divides the dataset into K equal-sized parts called folds, namely $D_1, D_2, ..., D_K$. Each fold $D_i$ is, in turn, treated as the testing set, with the remaining folds comprising the training set D $D_i = \cup_{j \neq i} D_j$. After training the model $M_i$ on D $D_i$, we assess its performance on the testing set $D_i$ to obtain the i-th estimate on $_i$. The expected value of the performance measure can then be estimated as

$$\hat{\mu} = E[\theta] = \frac{1}{K} \sum_{i=1}^{K} \theta_i \quad (2)$$

Usually K is chosen between 5 and 10. The testing set comprises a single point and the remaining data is used for training purposes. We enabled shuffling such that the folds are different each time.

We noticed in our case, the accuracy is the best for K=10. So the model is trained on 9 of the folds and the resulting model is validated on remaining part of the data. The performance measure reported by 10-fold cross-validation is then the average of the values computed in the loop.

We used StratifiedKFold cross validation iterator of sklearn package to split the data which is just a variation of KFold that returns stratified folds.

**Train Test Split:** This cross-validation algorithm splits arrays into random train and test subsets . This is a single split with a user defined split percentage. We mentioned the split percentage as 10% which implies 90% of the dataset is considered for training and the rest for test. We used train_test_split cross validation of sklearn package to split the data.

## 9. CLASSIFICATION

**NAIVE BAYES CLASSIFIER:**

For classification of tweets, we chose Naive Bayes because of its ease of use, simple design and good classification performance. Naive Bayes classifier is based on the model of probability. It focuses on the probability that a tweet belongs to a particular category C = $\{C_1, C_2, ..., C_K\}$. Here we use two categories (relevant and not-relevant). Each tweet is considered as a sequence of word tokens and each token will be labeled as relevant or not relevant. We calculate the posterior probability that the tweet belongs to either of the categories and chose the category with the highest probability.

The posterior probability of that a tweet $t_i$ belongs to a category $c_j$ can be calculated in formula:

$$p(c_j \mid t_i) = \frac{p(c_j)p(t_i)}{\sum_{k=1}^{|c|} p(c_k)p(t_i)} \tag{3}$$

$$Relevance(c_j, t_i) = log\frac{p(c_j \mid t_i)}{p(c_j^- \mid t_i)} \tag{4}$$

In order to evaluate the above we need to first calculate the probability of a category $c_j$ i.e. $p(c_j)$ and the probability of the tweet $t_i$ i.e. $p(t_i)$. $p(c_j)$ is given as the ratio of the number of tweets that fall into the category $c_j$ and the total number of tweets.

$$p(c_j) = \frac{n(c_j)}{n\left(\sum_{j=1}^{n} c_j\right)} \tag{5}$$

Here, $n(c_j)$ is the total number of tweets which belong to category $c_j$.

$p(t_i)$ can be evaluated based on all the words that make the tweet. We make an assumption that the words that make up the tweets are conditionally independent. Also we use a unigram, bigram, trigram language model. We get $p(t_i)$ as:

$$p(t_i) = \prod_{i=1}^{n} p(w_i \mid c_j) \tag{6}$$

where $p(w_i)$ is the probability that a word $w_i$ belongs to the category $c_j$.

The only parameters we will estimate are $p(c_j \mid t_i)$ $p(c_j^- \mid t_i)$ with the help of multivariate approach.The multinomial approach [4] specifies that a tweet is represented by the set of word occurrences from the tweet. A tweet is considered as a |V|- dimensional vector T=$(w_1, w_2, ..., w_{|V|})$ which is the result of |V| independent Bernoulli trials, where |V| is the vocabulary size $w_k$ is a binary variable representing the occurrence or non-occurrence of the k-th word in the vocabulary. The order of the words is lost, but the number of occurrences of each word in the tweet is captured. When calculating the probability of a tweet, one multiplies the probability of the words that occur. In the multinomial approach, the formula(3) is calculated as follows

$$\log\frac{P_n(t_i|c_j)}{P_n(t_i|c_j)} = \sum_{k=1,\omega\in t_i}^{|V|} TF_{ik}.\log\frac{P_n(\omega_k|c_j)}{P_n(\omega_k|\bar{c_j})} \tag{7}$$

We used MultinomialNB class in sklearn package for classification . A Laplace smoothing parameter of 0.1 is added to eliminate zeros. This is a smoothing algorithm that simply adds one to each count.

**OTHER CLASSIFIERS CONSIDERED - SVM:**

Another model we used is LinearSVC class from sklearn package for classification. LinearSVC (Linear Support Vector Classification) is based on Support Vector Model classifier. LinearSVC is similar to SVC(Support Vector Classifier) but is implemented in terms of liblinear which is a linear classifier for data with millions of instances and features.. A support vector machine is a classifier which classifies two classes by constructing a hyper plane in which the margin between the two classes will be maximum.SVMs are a generally applicable tool for machine learning. Suppose we are given training examples $x_i$, and the target values $y_i$ -1,1. SVM searches for a separating hyperplane, which separates positive and negative examples from each other with maximal margin, in other words, the distance of the decision surface and the closest example is maximal.

$$w^T x + b = 0 \tag{8}$$

The classification of an unseen test example x is based on the sign of $w^T x + b$. The binary classifier can be classified as

$$w^T x + b \geq 1 if y_i = +1 \tag{9}$$

$$w^T x + b \leq -1 if y_i = -1 \tag{10}$$

$K(x_i, x_j)$ denote a function that roughly speaking gives how similar two examples are. This is called a kernel-function. The decision function can be written as

$$f(x) = sign(\sum_{i=1}^{N} \alpha_i y_i.K(x_i, x_j) + b) \tag{11}$$

In general when dealing with Web collection data, SVM needs a sufficient number of positive and negative samples which need to be updated timely with increases in the size of the data set. This is a very costly process and with the given resources for the training data-set Naive Bayes seems to give better performance.In our implementation we classified data as the positive sample and consider rest of the data as negative sample. The accuracy of the system was comparatively less compared to Multinomial Naive Bayes. SVM implementation is more complex as it is time consuming to tune the internal parameters to get the best result and performance. We have chosen Naive Bayes over SVM as our mathematical model since the training time and processing time for the algorithm are comparatively less and it is fast, robust and simple to implement.

## 10. EVALUATION

We trained our system on a part of the human-provided labels (we will manually label the training set) and test the system on the remaining part. We evaluated our system by comparing outputs to the expected responses. We measured the below two aspects which are related to the sensitivity and the specificity of our system.

**Hit ratio/ precision rate (PR):** Precision rate measures the fraction of examples for which our system found something, and that something could be considered correct by humans. We will consider the output correct if it overlaps in at least one word with the given human label.

$$PR = \frac{num\ of\ rightly\ classified\ and\ retrieved\ data}{total\ retrieved\ data\ in\ the\ category} \tag{12}$$

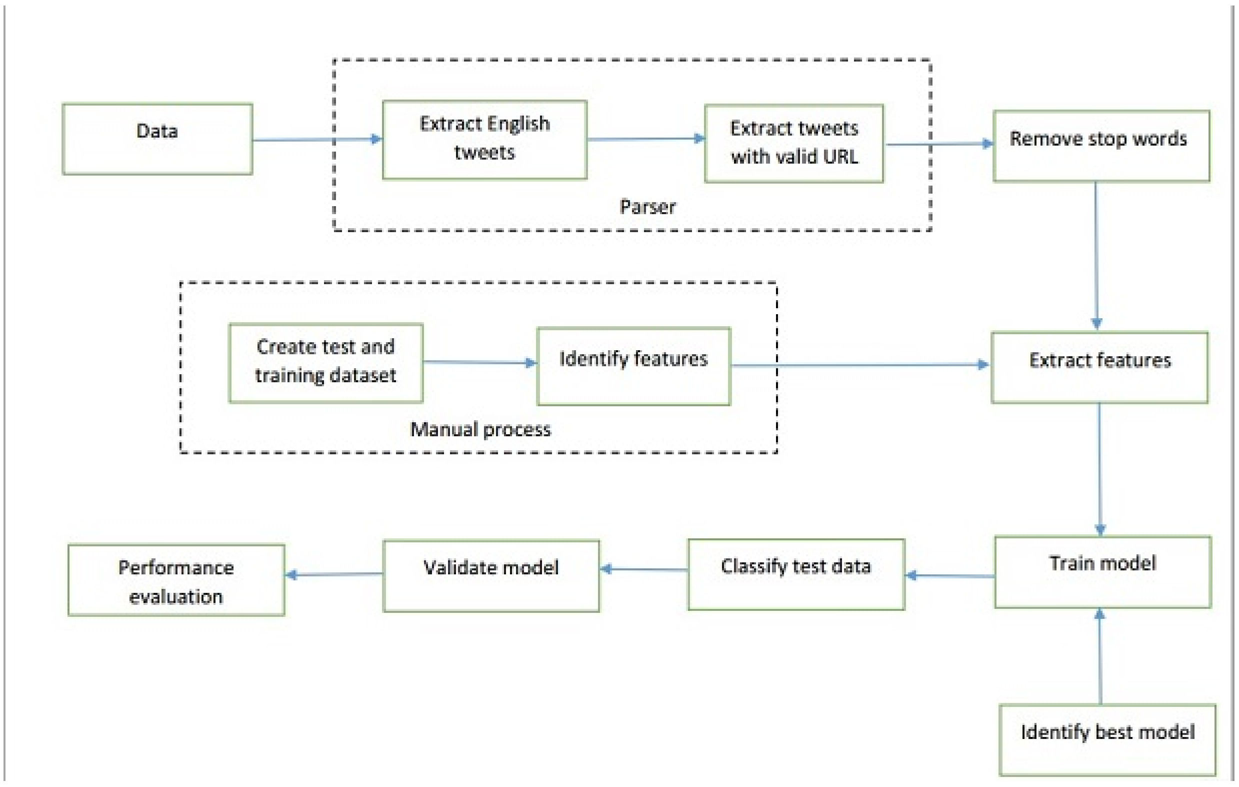


**Figure 1: A general framework of our classification model**

**Detection rate / recall rate (RR):** Recall rate will measure the fraction of examples in which humans found a relevant piece of information, and our system also found something.

$$RR = \frac{num\ of\ rightly\ classified\ and\ retrieved\ data}{rightly\ classified\ data\ in\ the\ category} \qquad (13)$$

We use Kfold cross validation for both models Figure[2,3].For Naive Bayes ,we notice that the performance increases steadily with the increase in k. It is the highest when k=10. This is because with the increase of K, the size of training data increases which means more information to the classifier when it is trained. Also, the size of test data decreases which means less data to be classified that is minimum error ratio. However this does not signify that the performance will continue to increase infinitely as k increases. With increase in the volume of training data, the classifier tends to overfit information which leads to more failures. So heuristics suggest that performance is optimal when k=10.
We have implemented the k-fold cross validation for the SVM classifier. It has been observed that SVM accuracy, precision, recall is less than the Naive Bayes.The less accuracy is due to the more number of false negative as in the learning data only the positive tweets has been classified and the remaining tweets are considered as a negative sample.
We also use train_test_split cross validation on both models (Multinomial Naive Bayes and Linear SVC)Figure[4,5]. For Naive Bayes classifier, we notice that precision is optimal when test size is 0.15 but recall decreases. This is a classical recall-precision trade-off where precision decreases while recall increases. We need to find a well-defined maximum which happens at test_size=0.20.We observe that for the train_test_split the accuracy of the SVM is approximately same as that of the Naive Bayes.

## 11. RELATED WORK

Twitter, as one of the most popular microblogging services, provides large quantities of fresh information including real time news, comments, conversation,pointless babble and advertisements.Twitter presents tweets in chronological order. The work of Duan et al. [2] proposes a new ranking strategy which uses not only the content relevance of a tweet, but also the account authority and tweet - specific features such as whether a URL link is included in the tweet .They used learning to rank algorithms to determine the best set of features with a series of experiments. It is demonstrated that whether a tweet contains URL or not , length of tweet and account authority are the best conjunction .

Cross-domain text classification aims to automatically train a precise text classifier for a target domain by using labeled text data from a related source domain. Most existing methods do not explore the duality of the marginal distribution of examples and the conditional distribution of class labels given labeled training examples in the source domain.In this paper [1],Bao proposes a model called Partially Supervised Cross-Collection LDA topic model (PSC-CLDA) for cross-domain learning with the purpose of addressing these two issues in a unified way. Experimental results on nine datasets show that their model outperforms two standard classifiers and four state-of-the art methods, which demonstrates the effectiveness of their proposed model.

A framework for summarizing and analyzing twitter feeds, [9]proposes a novel technique to analyze and summarize twitter data. The proposed method is light weight because it is an incremental sum-

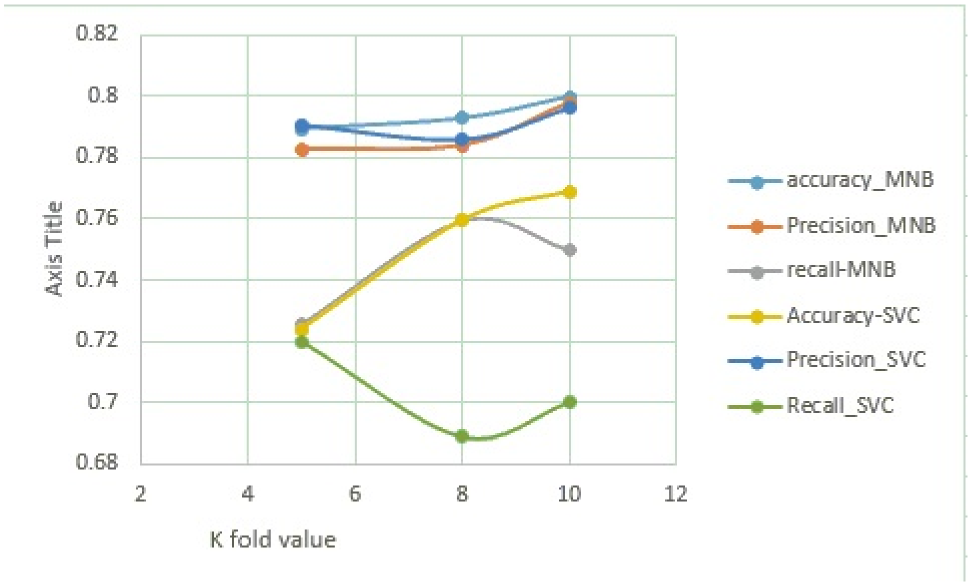
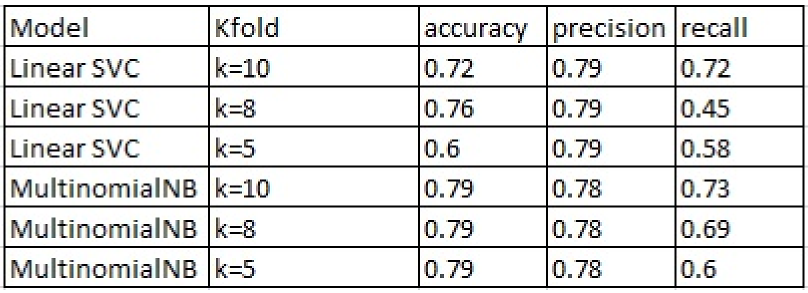


**Figure 2: CrossValidation KFold graph with k=5,8,10 for Linear SVC and Multinomial Naive Bayes models**



| Model | Kfold | accuracy | precision | recall |
|---|---|---|---|---|
| Linear SVC | k=10 | 0.72 | 0.79 | 0.72 |
| Linear SVC | k=8 | 0.76 | 0.79 | 0.45 |
| Linear SVC | k=5 | 0.6 | 0.79 | 0.58 |
| MultinomialNB | k=10 | 0.79 | 0.78 | 0.73 |
| MultinomialNB | k=8 | 0.79 | 0.78 | 0.69 |
| MultinomialNB | k=5 | 0.79 | 0.78 | 0.6 |

**Figure 3: CrossValidation KFold data table with k=5,8,10 for Linear SVC and Multinomial Naive Bayes models**

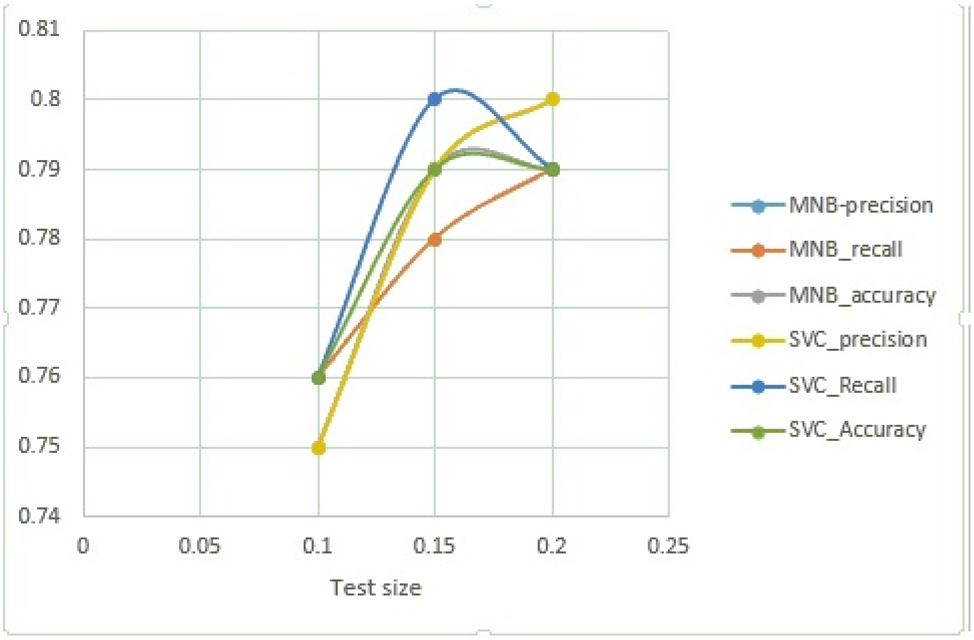
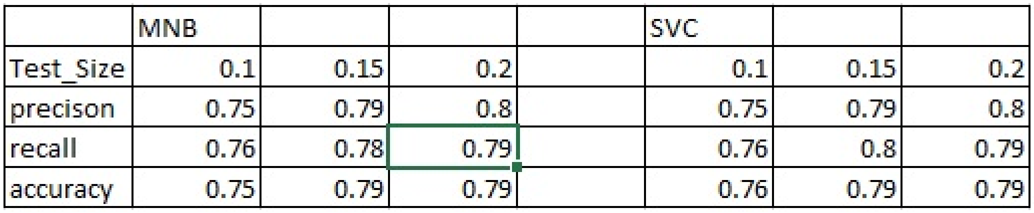


**Figure 4: Train_Test_Split graph for Linear SVC and Multinomial Naive Bayes models**



|  | MNB |  |  |  | SVC |  |  |
|---|---|---|---|---|---|---|---|
| Test_Size | 0.1 | 0.15 | 0.2 |  | 0.1 | 0.15 | 0.2 |
| precison | 0.75 | 0.79 | 0.8 |  | 0.75 | 0.79 | 0.8 |
| recall | 0.76 | 0.78 | 0.79 |  | 0.76 | 0.8 | 0.79 |
| accuracy | 0.75 | 0.79 | 0.79 |  | 0.76 | 0.79 | 0.79 |

**Figure 5: Train_Test_Split table for Linear SVC and Multinomial Naive Bayes models**

mary construction which is efficient. It also enables low reconstruction error.

Yiming and Liu [10] represented a controlled study with statistical significance tests on five text categorization methods: the Support Vector Machines (SVM), a k-Nearest Neighbor (kNN) classifier, a neural network (NNet) approach, the Linear Least-squares Fit (LLSF) mapping and a Naive Bayes (NB) classifier.The paper focuses on the robustness of these methods in dealing with a skewed category distribution, and their performance as function of the training set category frequency.They show that SVM, kNN and LLSF significantly outperform NNet and NB when the number of positive training instances per category are small, and that all the methods perform comparably when the categories are sufficiently common.

By analogy with TF-IDF model, Rousseau [5] proposed a new technique TW-IDF model which assigns term weights (TW) rather than term frequency. They capture the relationship between terms using unweighted graphs and extract meaningful term weights. We plan to add term weights to our list of words in feature list to increase accuracy of our classifier and we feel this model might lead us to the right direction.

Lan, [3] represents an empirical study of top-k learning to rank training in this paper . The challenge in learning to rank algorithms is getting reliable training data. So top k learning algorithm proposes to utilize top-k ground truth for training where k is usually small. The underlying assumption is that training on top-k grade truth is as good as full-order ranking list. They study whether this underlying assumption of top-k learning ground-truth is sufficient for training holds.

In the work done by Shang. [7] the major problem of text classification i.e.high dimensionality of the feature space is addressed. They proposed a novel Gini index algorithm to reduce the high dimensionality of the feature space. A new measure function of Gini index is constructed and made to fit text categorization. The results of experiments show that the improvements of Gini index behave better than other methods of feature selection.

## 12. PROBLEMS FACED

The main issue was to correctly identify the features while sampling the data for good precision and accuracy.We have to manually label the data which was time consuming. The challenge we are trying to address is to add the correct stop words. We were using the standard English stop word dataset. However we noticed that the parts or the URLs are becoming features. We are trying to modify our stop-word list to have a better accuracy, precision and recall.

## 13. CONCLUSION

To summarize, we proposed the following contributions: (1) classify tweets-with-URLs based on tweet text.(2) unshorten those URLs and extract text from the corresponding web-pages (3) index all the collected data in data in Solr (4) build a simple interface so that the index data can be searched.
We have completed step 1 which is approximately 30% of the project and we plan to complete rest of the proposed steps by Dec 1.

## 14. SOFTWARES/PACKAGES USED

Python: version 2.7
Latex: version 2e
Python packages used:
NLTK : for stop words
Sklearn : for classification
Numpy : for converting text to array
Pandas: for reading .csv file

## 15. REFERENCES

[1] Y. Bao, N. Collier, and A. Datta. A partially supervised cross-collection topic model for cross-domain text classification. In *Proceedings of the 22nd ACM international conference on Conference on information & knowledge management*, pages 239–248. ACM, 2013.

[2] Y. Duan, L. Jiang, T. Qin, M. Zhou, and H.-Y. Shum. An empirical study on learning to rank of tweets. In *Proceedings of the 23rd International Conference on Computational Linguistics*, COLING '10, pages 295–303, Stroudsburg, PA, USA, 2010. Association for Computational Linguistics.

[3] Y. Lan, S. Niu, J. Guo, and X. Cheng. Is top-k sufficient for ranking? In *Proceedings of the 22nd ACM international conference on Conference on information & knowledge management*, pages 1261–1270. ACM, 2013.

[4] K. Nigam, A. K. McCallum, S. Thrun, and T. Mitchell. Text classification from labeled and unlabeled documents using em. *Machine learning*, 39(2-3):103–134, 2000.

[5] F. Rousseau and M. Vazirgiannis. Graph-of-word and tw-idf: new approach to ad hoc ir. In *Proceedings of the 22nd ACM international conference on Conference on information & knowledge management*, pages 59–68. ACM, 2013.

[6] T. Sakaki, M. Okazaki, and Y. Matsuo. Earthquake shakes twitter users: real-time event detection by social sensors. In *Proceedings of the 19th international conference on World wide web*, pages 851–860. ACM, 2010.

[7] W. Shang, H. Huang, H. Zhu, Y. Lin, Y. Qu, and Z. Wang. A novel feature selection algorithm for text categorization. *Expert Systems with Applications*, 33(1):1–5, 2007.

[8] S. Vieweg, A. L. Hughes, K. Starbird, and L. Palen. Microblogging during two natural hazards events: What twitter may contribute to situational awareness. In *Proceedings of the SIGCHI Conference on Human Factors in*

*Computing Systems*, CHI '10, pages 1079–1088, New York, NY, USA, 2010. ACM.

[9] X. Yang, A. Ghoting, Y. Ruan, and S. Parthasarathy. A framework for summarizing and analyzing twitter feeds. In *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 370–378. ACM, 2012.

[10] Y. Yang and X. Liu. A re-examination of text categorization methods. In *Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval*, pages 42–49. ACM, 1999.