International Refereed Journal of Engineering and Science (IRJES)
ISSN (Online) 2319-183X, (Print) 2319-1821
Volume 7, Issue 1 (January 2017), PP. 01-07

Real-TimeInformation Filtering on Classification of Trending Topics in Twitter

Qi Zhu
Department of Computer Science, University of Houston-Victoria, USA
Corresponding Author: Qi Zhu

Abstract: Twitter is an online news and network service where people can post and interact via messages, which has been proved to be the largest source of breaking news during 2016 U.S. presidential election. A novel way of real-time information filtering through text categorization of tweets is proposed in this paper to classify the trending topics, using a combination of short text aggregation and the multinomial Naïve Bayes classifier.

Date of Submission: 05-01-2018 Date of acceptance: 20-01-2018

I. INTRODUCTION

An excellent source of data for research on short text classification exists within Twitter, anapplication for sharing information with the public in the form of 140-charater long messages, called Tweets. The semantic structure of the messages, pose several problems for developers and researchers alike, including data sparseness, the lack of informationor context, slang, misspelled words, and acronym resolution. Currently within the Twitter, there is a feature which broadcasts the trending topics from around the world. Presently, there lacks an efficient way to contextualize the trending topics to understand if they are important to the user. In modern times information filtering has become a topic of great debate, for example, it is not unheard of to have over 400 channels within a single cable package, according to Apple, iTunes has over 43 million songs, with the continuous prevalence and ubiquity of mobile devices, humans are receiving an extraordinary amount of information. To effectively handle the excessive data, information filtering is now a necessity. In this paper,we propose a novel way of real-timeinformation filtering throughtext categorization of tweets,using a combination ofshort text aggregation and the multinomial Naïve Bayesclassifier.

II. LITERATURE REVIEW

Over the past several years, there has been particular interest in the research of short text classification, especially within the realm of micro-blogging, reviews, E-mails, and SMStext messages. While extensive research has and is continuing to be done in short text classification, there still exist a need for applied methods within the realm of information filtering. The following papers have all recently contributed to the advancement of short text classification, and dealing with the multitude of problems within this subject.

Short Text Classification In Twitter To Improve Information Filtering

The method presented by Sriram et al., in their 2010 paper, builds upon previous text categorization work done in the field of short text categorization. Some of the major differences are the focus on the author, intent, and historical user data. In this method of categorization, categories are predetermined, and based upon features present, tweets can fall into any of the 5 classes, including, News (N), Events (E), Opinions (O), Deals (D), and Private Messages (PM) (2010). When choosing features for categorization, a greedy strategy was employed to determine the feature set. Eight features were extracted, which consisted of one nominal attribute, the author of the tweet, and seven binary features (yes/no, 0/1, true/false) including, presence of shortening of words and slangs, time-event phrases, opinioned words, emphasis on words, currency and percentage signs, "@username" at the beginning of the tweet, and "@username" within the tweet (Sriram et al., 2010).

The primary feature used to classify text in this method is authorship, understanding the relationship between author and short text intent is a very complex task. For instance, individual authors can take on many different forms over the course of a micro blog, at one point they can be reporters, spreading local news to the masses, at other times event coordinators, promoting the time and location of an occasion, or even opinion authors, giving the perspective on public matters. In order to gain further insight in authorship, the text is mined for keywords, giving comprehension of author intent (Sriram et al., 2010). This method implemented by Sriram et al., produced an increase in accuracy over Bag-of-Words by 32.1%.

Discovering Context: Classifying Tweets through a Semantic Transform Based on Wikipedia

A method proposed by Genc et al., continues research done in short text classification, where outside knowledge bases are used to find context, and ultimately classify text. In research done by Stone et al., Wikipedia was used as training corpus to improve the ability of statistical methods to discover meanings of short texts (Stone, 2010). Michelson and Macskassy also used Wikipedia, and this is the model that was primarily expounded upon by Genc et al. In their model, tweets are transformed into a word set, all stop words are removed from this word set, then each individual word is mapped to Wikipedia pages. Then a list of candidate pages for the tweet is found by aggregating each page associated with each word of the word set. A score is computed for each candidate page by counting the number of occurrences of the words in the word set. The page with the highest score is selected as the associated Wikipedia page for the tweet (Genc, 2011). Since categories of Wikipedia pages are known due to their network structure, this label can then be applied to the tweets. Tweets can therefore, be labeled and or clustered with similar tweets (Genc, 2011). This technique was compared with two other methods, String Edit Distance (SED), and Latent Semantic Analysis (LSA).The accuracy results of classifying tweets with known categories, shows this technique outperforms SED by an average of 40% and outperforms LSA by an average of 13.33%. When the same test was compared with noisy data added in, the proposed technique outperformed LSA by an average of 37.67% (Genc, 2011).

Topic Classification in Social Media using Metadata from Hyperlinked Objects

Another approach introduced by Berendt et al., and later improved upon by Kinsella et al., is the use of metadata to aid in the classification of short text documents. Over the last decade there has been a dramatic increase in the use of tags within web documents. Tags can take on a variety of forms, including HTML, XML, as well as Open Graph Protocol and other content specific languages. These tags provide an additional source of information about the textual data. Another commonality amongst short text communications is the use of hyperlinks. Hyperlinks are use to direct readers to more in depth articles or webpages in relation to the short text.In Kinsella's et al. approach, metadata tags, hyperlinks, and the original text, are used to classify message board postings into discrete categories. In order to compare the usefulness as sources of features for topic classification, each post is derived into four different bag-of-words representations. C-L denotes the original content of the post, with all hyperlinks removed, while C denotes the full original content with hyperlinks intact. M denotes the external metadata retrieved from the hyperlinks of the post. C+M denotes the combination of C and M for a given post, i.e. the full original content plus the external metadata retrieved from its hyperlinks (Kinsella, 2011).

A Naïve Bayes classifier was implemented in Weka to classify the postings. To create the Bag-of-Word representation, each posting received the following transformations; all text is changed to lower-case, all non-alphabetic characters are replaced with spaces. Stopwords are removed, and TF-IDF and document length normalization are applied. A ten-fold cross validation was used to evaluate classifier performance on each type of document representation (Kinsella, 2011). The average F-score for each classifier was .765 for C-L, .815 for C, .837 for M, and .883 for C+M. The experiments by Kinsella et al.,revealed that external metadata has better descriptive power for topic classification than the original posts, and that a combination of both gives best results (Kinsella, 2011).

Twitter Trending Topic Classification

Lee et al. did another study that closely relates to the methodology proposed in this paper, in their research, they suggested a model for classification of Twitter's trending topics using a Naïve Bayes Multinomial Classifier. Their goal was to aid users searching for information on Twitter, by looking at only a smaller subset of trending topics by classifying topics into general classes (e.g., sports, politics, books) for easier retrieval of information (Lee, 2011). To address this problem, they defined 18 general classes: arts & design, books, business, charity & deals, fashion, food & drink, health, holidays & dates, humor, music, politics, religion, science, sports, technology, TV & movies, other news, and other (Lee, 2011). To classify trending topics into these predefined classes, they proposed two approaches, Bag-of-Words text classification, and using social network information. For the social network information approach, topic tweets were weighted based upon the user, using the Twitter Friend-Follower network. A specific user's tweet has more influence if the sum of the influence of their followers is high (Lee, 2011).

In their experiments, they initially used a random sample of 1000 topics, fromthere; the data set was narrowed down to 768 topics. These topics were then hand labeled by two humans, base upon the 18 predefined categories (Lee, 2011).Using WEKA and SPSS modeler, to build their models,in all experiments, 10- fold cross-validation was used to evaluate the classification accuracy. The ZeroR classifier was used to get a baseline accuracy, which simply predicts the majority class. The text based classification approach using the Naïve Bayes Multinomial classifier achieved an accuracy of 65.36%. The alternate approach, network based classification, used a variety of classification techniques to test performance, C5.0 decision tree classifier gives

best classification accuracy (70.96%), followed by k-Nearest Neighbor (63.28%), Support Vector Machine (54.349%), and then Logistic Regression (53.457%) (Lee, 2011).

III. METHODOLOGY

Twitter as a Source of Data

In this study, the proposed methodology is to use the R programming language in conjunction WEKA and several other packages, to acquire tweets, preprocess the data, and ultimately categorize the tweets. The first step is to use Twitter's API, which will allowthe 10 currently trending topics to be obtained. These trending topics are then placed into a table. It is key to only gather currently or recently trending topics. What was discovered over the course of this study, is the misuse of trending topics for promotion of non-related products or information; It is only the currently trending topics which have the actual tweets that are related to the topic when gathered in the next step.

The next step would involve looping through all trending topics, and using the Twitter API to search for the tweets related to the trending topics, retrieving up to300 tweets related toeach topic. Within this loop, tweets are retrieved and stored in a vector, representing a bag of words format. After the data is obtained, stop words are removed, and a custom cleaning function is applied to the tweets, eliminating "@names," the abbreviation "rt," newlines, punctuations, numbers, tabs, links, and extra whitespaces. In addition, at thisstage, any non-ASCII characters are also removed from the text. A snippet of this code can be viewed below.

```
#custom data cleansing function
clean = function(x){
  #remove non-graphical characters
  x = gsub("[^[:graph:]]", "", x)
  #tolower
  x = tolower(x)
  #remove rt
  x = gsub("rt", "", x)
  #remove at
  x = gsub("@\\w+", "", x)
  #remove end line
  x = gsub("\n", " ", x)
  #remove punctuation
  x = gsub("[[:punct:]]", "", x)
  #   #remove numbers
  #   x = gsub("[[:digit:]]", "", x)
  #remove links http
  x = gsub("http\\w+", "", x)
  #remove tabs
  x = gsub("[ |\t]{2,}", "", x)
  #remove blank spaces at the beginning
  x = gsub("^ ", "", x)
  #remove blank spaces at the end
  x = gsub(" $", "", x)
  return(x)
}
```

Figure 1. An Example in R Program

After the cleaning of the data each word in the vector is compared to predefined list of words commonly related to the one of the categories. The count of each trending topic is summed and ranked. Based on that ranking, theTrending Topic is assigned a preliminary label. This code is illustrated below.

```
######### Data preprocessing and Transformation #########

#aggregation of tweets and conversion to bag of words
  trend_bow <- paste(trend$text, collapse = " ")
  trend_bow <- unlist(strsplit(trend_bow, " "))
  trend_bow <- clean(trend_bow)
  trend_bow <- removeWords(trend_bow, c(stopwords("english")))
  trend_bow <- trend_bow[trend_bow != ""]
  trend_words <- data.frame(table(trend_bow))
  trend_words <- trend_words[which(trend_words$Freq > 10),]
#Order by Frequency in descending order, extract Top 100 words, put in list
  trend_words <- trend_words[order(-trend_words$Freq),]
#   trend_words <- trend_words[1:1000,]
  trend_wFreq <- unlist(list(trend_words$Freq))
  trend_words <- unlist(list(trend_words$trend_bow))

  e = 0
  s = 0
  n = 0
  for (word in 1:length(trend_words))
  {
      ifelse(trend_words[word] %in% EntertainmentWords, e <- e + as.numeric(trend_wFreq[word]), NA)
      ifelse(trend_words[word] %in% NewsWords, n <- n + as.numeric(trend_wFreq[word]), NA)
      ifelse(trend_words[word] %in% SportsWords, s <- s + as.numeric(trend_wFreq[word]), NA)
  }

  h <- max(c(e, n, s))
  ifelse( (h == e) && ( (e/length(trend_words)) > .25 ), trends[i,3] <- "Entertainment", NA )
  ifelse( (h == n) && ( (n/length(NewsWords)) > .25 ), trends[i,3] <- "News", NA )
  ifelse( (h == s) && ( (s/length(SportsWords)) > .25 ), trends[i,3] <- "Sports", NA)
  ifelse(trends[i,3] == "None" , trends[i,3] <- "Other" , NA)

  trend_bow <- paste(trend_bow, collapse = " ")
```

Figure 2. An Example of Trending Topics in R Program

Following the preprocessing of the data, the multinomial Naïve Bayes algorithm is applied.The multinomial Naïve Bayes is a specialized form of the Naïve Bayes, the main difference lies in its ability to capture word frequency from within documents.

Multinomial Naïve Bayes

Multinomial Naïve Bayes (MNB) models the representation of text in a document as a polynomial function. The document is considered anarrangement of words and there is an assumption that the wordslocation is independent of every other, this is the naïve part. For categorization, we assume that there are a permanent number of classes, each having apermanentnumber of parameters. The parameter vector for a class c is $\vec{\theta}_c = \{\theta_{c1},\theta_{c2},...,\theta_{cn}\}$, where n is the size of the vocabulary. The likelihood of a document can be represented by the appearance of words, giving us the following equation,

$$p(d|\vec{\theta}_c) = \frac{(\sum_i f_i)!}{\prod_i f_i!} \prod_i (\theta_{ci})^{f_i},$$

where f_i is the frequency of word i in document d. By assigning a prior distribution over all sets of categories, $p(\vec{\theta}_c)$, we can arrive at the "min-error classified rule" (Duda & Hart, 1973) which selects the class with the largest posterior probability,

$$l(d) = \text{argmax}_c \left[\log p(\vec{\theta}_c) + \sum_i f_i \log \theta_{ci} \right],$$

$$= \text{argmax}_c \left[b_c + \sum_i f_i w_{ci} \right],$$

where b_c is the limit and w_{ci} is the category c weight for word i. In0/1 classification, the limit is stated by setting the differences between the positive and negative class parameters equal to zero,

$$(b_+ - b_-) + \sum_i f_i \, (w_{+i} - w_{-i}) = 0.$$

This equation is similar to equations in (linear) Support-Vector-Machine, logistic regression, and Neural Networks (MLP). MNB'simplementation results from how it chooses the b_c and w_{ci}.

When classifying data using supervised methods, there is usually a predetermined amount of classes. All classification methods use a feature set or parameters to label each object, where theparameters should containpertinent information. The parameters are generated using training data. This done by using the Dirichlet distribution as a prior distribution, which gives us the number of times the word appears in the data, *i*, divided by the number of word occurrences in the class, *c*. For *i*, a prior adds in αi fictional existences (additive smoothing) so that the approximation is a "smooth version" of the maximum likelihood estimate,

$$\hat{\theta}_{ci} = \frac{N_{ci} + \alpha_i}{N_c + \alpha},$$

where $\alpha$is the summation of αi. αiequals1 for all words, which sets a minimum of 1 occurrence for everyword. This is the basis for thisMultinomial Naïve Bayes classifier,

$$l_{\text{MNB}}(d) = \text{argmax}_c \left[\log \hat{p}(\theta_c) + \sum_i f_i \log \frac{N_{ci} + \alpha_i}{N_c + \alpha} \right]$$

Where this model differs from the methods proposed Lee et al, is in the use of Term Frequency-Inverse Document Frequency, which is the weighting factor used to determine the importance of a word given documents. The following statistical weighting formulas for Term Frequency and Inverse Document Frequency are shown as:

$$1 + \log f_{t,d},$$

$$log\left(1 + \frac{N}{n_t}\right).$$

The more words appear in a single document, the more weight it holds for the specific document type, however, as the word appears in more documents its weight will decrease.

Where this model differs from Lee et al. is in the use of td-idf weights, instead of using raw term frequencies and in the use of additive smoothing, which allow for the assignment of non-zero probabilities to words that do not appear in the training data.

IV. EXPERIMENT RESULTS

To assess the advantage that Multinomial Naive Bayes offers over several other options including Neural Networks, Decision Trees, and Latent Semantic Analysis, etc., the proposed method was tested on the data acquired in the experiment. At the onset of the Data Collection and Preprocessing phase, over 220 trending topics were collected, eachhaving 300 related tweets, compromised on average of 21words. Thus, the initial dataset contained about 1.4 million words. During preprocessing the custom transformations as detailed above were implemented reducing the final size down for the training phase.During the training phase a sample of 160 trending topics was retained. For each topic, 300 related tweets were sampled, on average containing 14 words. Therefore, the trained model was based on a corpus of over 672,000 words related to 160 trending topics, taking 1.5 seconds to build.

After building the model, it was tested against a testing dataset composed of 30 trending topics, with the following category composition; 13 Sports, 7 Entertainment, 6 News, and 4 Other. The overall accuracy of this model showed 73.33% accuracy, classifying 22/30 topics correctly. The overall weighted average precision and recall, were .73 and .733 respectively. This resulted in an average F-Measure of .715 and a ROC Area (or AUC) of .893.

```
=== Summary ===

Correctly Classified Instances          22               73.3333 %
Incorrectly Classified Instances         8               26.6667 %
Kappa statistic                          0.6066
Mean absolute error                      0.1427
Root mean squared error                  0.3534
Relative absolute error                 39.7195 %
Root relative squared error             83.8722 %
Total Number of Instances               30

=== Detailed Accuracy By Class ===

               TP Rate   FP Rate   Precision   Recall  F-Measure   ROC
Area  Class
                 0.429     0.043      0.75      0.429     0.545
0.814    Entertainment
                 0.667     0.125      0.571     0.667     0.615
0.799    News
                 0.5       0.038      0.667     0.5       0.571
0.971    Other
                 1         0.176      0.813     1         0.897
0.955    Sports
Weighted Avg.    0.733     0.117      0.73      0.733     0.715
0.893
```

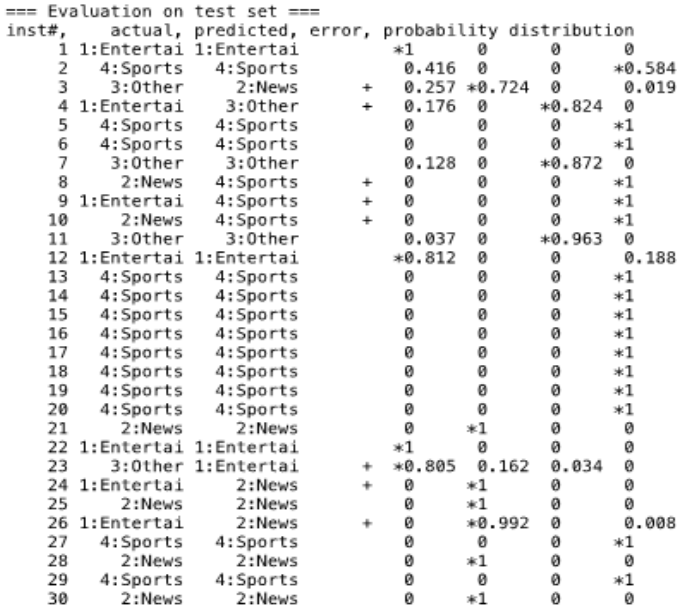
Figure 3. Model Classification Result

```
=== Confusion Matrix ===

  a  b  c  d   <-- classified as
  3  2  1  1 |  a = Entertainment
  0  4  0  2 |  b = News
  1  1  2  0 |  c = Other
  0  0  0 13 |  d = Sports
```

Figure 4. Model Classification Confusion Matrix

Upon further evaluation of the running results it is clear that the model has an accurate representation of the category “Sports”, with a precision of .813 and a recall of 1. Where this model struggled, was in its ability to classify “Entertainment”, and “News” text, sometimes confusing the two. “Entertainment” produced a precision of .75, but only had a recall of .429, while “News” had a recall and precision of .571, and .667 respectively.



```
=== Evaluation on test set ===
inst#,    actual, predicted, error, probability distribution
     1 1:Entertai 1:Entertai        *1      0      0      0
     2   4:Sports   4:Sports          0.416  0      0     *0.584
     3    3:Other     2:News      +   0.257 *0.724  0      0.019
     4 1:Entertai    3:Other     +   0.176  0     *0.824  0
     5   4:Sports   4:Sports          0      0      0     *1
     6   4:Sports   4:Sports          0      0      0     *1
     7    3:Other    3:Other          0.128  0     *0.872  0
     8     2:News   4:Sports      +   0      0      0     *1
     9 1:Entertai   4:Sports      +   0      0      0     *1
    10     2:News   4:Sports      +   0      0      0     *1
    11    3:Other    3:Other          0.037  0     *0.963  0
    12 1:Entertai 1:Entertai        *0.812  0      0      0.188
    13   4:Sports   4:Sports          0      0      0     *1
    14   4:Sports   4:Sports          0      0      0     *1
    15   4:Sports   4:Sports          0      0      0     *1
    16   4:Sports   4:Sports          0      0      0     *1
    17   4:Sports   4:Sports          0      0      0     *1
    18   4:Sports   4:Sports          0      0      0     *1
    19   4:Sports   4:Sports          0      0      0     *1
    20   4:Sports   4:Sports          0      0      0     *1
    21     2:News     2:News          0     *1      0      0
    22 1:Entertai 1:Entertai        *1      0      0      0
    23    3:Other 1:Entertai      +  *0.805  0.162  0.034  0
    24 1:Entertai     2:News      +   0     *1      0      0
    25     2:News     2:News          0     *1      0      0
    26 1:Entertai     2:News      +   0     *0.992  0      0.008
    27   4:Sports   4:Sports          0      0      0     *1
    28     2:News     2:News          0     *1      0      0
    29   4:Sports   4:Sports          0      0      0     *1
    30     2:News     2:News          0     *1      0      0
```

Figure 5. Model Classification Evaluation

V. DISCUSSION

When compared to similar research, the model proposed here performed resoundingly well. Over the method results attained by Lee et al., in 2011, (C5.0 decision tree (70.96%), k-Nearest Neighbor (63.28%), Support Vector Machine (54.349%), and then Logistic Regression (53.457%) (Lee, 2011).), the MNB model increased accuracies over C5.0 by 3.4%, k-Nearest Neighbor by 15.88%, Support Vector Machine by 34.92%, and Logistic Regression 37.18%. When compared to the methods proposed by Genc et al., the MNB model increased accuracies over String Edit Distance by 57.12%, was even with Latent Semantic Analysis, and fell short of the Wikipedia Search Method by 16%. The model also fell short of Kinsella et al., method when comparing average F-Scores. While Kinsella achieved an average F-Score of .883, MNB was only able to achieve an average F-Score of .715. Additional information worth noting is that for these systems to be used in the actual Twitter application, they would have to be able to classify tweets in real-time. This is something that MNB excels at. It only took 1.5 seconds to build model and classify 30 trending topics. When compared to other methods, Neural Networks being at the most extreme end, can take up to 4 hours to accurately build a model capable of performing this same task.

In addition, some of the better performing models, like the one produced by Genc et al., all have external information gathering to increase the attributes for each trending topic; In this instance, Genc used Wikipedia. I believe that the model proposed here could also benefit from the external information and see a boost in accuracies. This is potentially an area of future research. However, it has been shown here, that short text classification of real-time information is possible and has the potential to change the functionality of Twitter for future users.

REFERENCES

[1]. Becker, M. and Gravano, L. "Beyond trending topics: Real-world event identification on twitter," in Proceedings of AAAI, 2011.

[2]. Berendt, B., Hanser, C.: Tags are not metadata, but "just more content"–to some people. In: Proceedings of ICWSM, 2007.

[3]. Genc, Y., Sakamoto, Y., and Nickerson, J. "Discovering context: Classifying tweets through a semantic transform based on Wikipedia," in Proceedings of HCI International, 2011.

[4]. Gentry, J. "Package: TwitteR, R based Twitter client." 2014.

[5]. Kinsella, S., Passant, A., and Breslin, J. "Topic classification in social media using metadata from hyperlinked objects," in Proceedings of the 33rd European conference on Advances in information retrieval, 2011, pp. 201–206.

[6]. Lee K, Palsetia D, Narayanan R, Patwary MMA, Agrawal A, Choudhary A Twitter trending topic classification. 11th IEEE international conference on data mining workshops (ICDMW), 2011.

[7]. Linoff, G., Berry, M. "Data Mining Techniques." Wiley Publishing. Third Edition. 2011.
[8]. Michelson, M. and Macskassy, S. A.: Discovering users' topics of interest on twitter: A first look. Proceedings of the Workshop on Analytics for Noisy,
[9]. Unstructured Text Data. (2010)
[10]. Sriram, B., Fuhry, D., Demir, E., Ferhatosmanoglu, H. and Demirbas, M. "Short text classification in twitter to improve information filtering," in Proceeding of the 33rd international ACM SIGIR conference on Research and development in information retrieval, 2010, pp. 841–842.
[11]. Stone, B., Dennis, S., and Kwantes, P. J.: Comparing Methods for Single Paragraph Similarity Analysis. Topics in Cognitive Science. Wiley Online Library, 2010.

Qi Zhu. "Real-TimeInformation Filtering on Classification of Trending Topics in Twitter." International Refereed Journal of Engineering and Science (IRJES), vol. 07, no. 01, 2018, pp. 01–07.