Twitter Trolls and the Tweeters who Love them

Anonymous

1 Introduction

The recent troll detection system are mainly based on Recurrent Neural Networks (RNN) with bidirectional Long Short-Term Memory (LSTM) [1], Support Vector Machine (SVM) with sentiment analysis [2], etc.

The goal of this report is to demonstrate the knowledge about how to classify the troll tweets based on their text automatically. Each of the tweets could be classified as one of the three classes: "LeftTroll", "RightTroll", or "OtherTroll", where the left and right troll users corresponds to the people who support left-wing or right-wing in US politics.

In this report, two knowledge problems will be raised, and a few machine learning methods will be discussed and applied over a dataset comprising a number of "troll" tweets to gain knowledge about the problems.

1.1 Dataset

The dataset used in this report involves 223K tweets published by 175 users, randomly chosen from 3 million Russian Troll tweets. [3] The tweets have been preprocessed by removing the non-English alphabets characters, and lower-casing all of the characters.

The "medium" dataset will be used for applying the machine learning methods and gaining knowledge in the interest of speed. In the dataset, the "best50" features are used as the initial features, where the terms with the highest Mutual Information or Chi-Square values for each class were selected.

1.2 Knowledge Problem

There could be many interesting knowledge problems about automatically identify trolls. However, this paper will only be focusing on two knowledge problems mainly based on the Naive Bayes (NB) method:

1. Is NB useful for classifying the trolls? Why or why not?
2. How to improve the features for NB for identifying trolls?

2 NB usefulness

2.1 Baseline algorithms

The usefulness of NB could be demonstrated by comparing with two baseline algorithms: Zero-R and One-R. For this dataset, Zero-R is classifying all instances to the "OtherTroll" class, since it is the majority class in the training set, and One-R is using the rules associated with "user-id". As shown in Table 1 below, NB has a much higher accuracy than both of the baseline algorithms.

Evaluation Metric	Accuracy(%)
Zero-R	35.785
One-R	25.5526
NB	60.1167

Table 1: Evaluating Zero-R, One-R, and NB with development set

2.2 Benchmark algorithms

Comparing with a few benchmark algorithms: Support Vector Machine (SVM), Decision Tree (DT), and Logistic Regression (LR), NB does not perform as good as these methods. The lower accuracy may be because of the "NB Conditional Independence Assumption", which assumes the independence between the attributes. This issue could be solved by using "Naive Bayes with Single Machine Gaussian", which is more robust for the attributes dependent on each other. [4]

However, NB still has a very competitive accuracy. Moreover, it is extremely fast to make predictions and easy to change probabilities when new data becomes available.

Model	Accuracy(%)
SVM	60.1666
LR	62.4746
NB	60.1167

Table 2: Evaluating SVM, LR, and NB with development set

2.3 NB for three classes

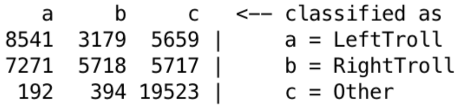
As shown in Table 3 below, the NB model is more effective on the "OtherTroll" class, which has the highest F-Measure (the harmonic mean of precision and recall) among all three classes.

The three classes have similar precision, but "LeftTroll" and "RightTroll" have lower recalls, which indicates that many "LeftTroll" and "RightTroll" instances were classified incorrectly.

Class	Precision	Recall	F-Measure
Left	0.534	0.491	0.512
Right	0.615	0.306	0.408
Other	0.632	0.971	0.765

Table 3: NB models with Numeric and Binary attributes

By analysing the confusion matrix below, it was found that the "LeftTroll" tweets tends to be incorrectly classified as "OtherTroll" more than "RightTroll", and more than one third of the "RightTroll" tweets were misclassified as "LeftTroll".



```
    a     b     c   <-- classified as
 8541  3179  5659 |     a = LeftTroll
 7271  5718  5717 |     b = RightTroll
  192   394 19523 |     c = Other
```

Figure 1: Confusion matrix of the NB model

Here are a few examples of incorrectly classified tweets from two categories:

1. **C1: "LeftTroll" as "OtherTroll"**:

   ID-2125: Police in the US aren't racists? Yeah right. ÔøΩ https://t.co/PfG3hJAaHR

   ID-2236: Stop this Police Madness! https://t.co/ESGJf6otNu

   ID-2267: Black teenage girl sues white police officer for brutal assault in car park #BTPVideo https://t.co/MGn2HePHBf https://t.co/AgbwT8WPcK

2. **C2: "RightTroll" as "LeftTroll"**:

   ID-257: So the only good people in this world are black transexual lesbians in a wheelchair. @steve0423 Check this #libtard https://t.co/tkOEWi5qvU

   ID-2269: Do you think if there are more white people than black it means racism? #OscarHasNoColor

   ID-2315: '@KdubSoSolid Why is dating black supermodels gross? I strongly disagree, sir!'

For the tweet examples in the first category above, the three "LeftTroll" tweets were classified as "OtherTroll". One of the reasons may be that the term "police" has a higher mean value in the training set (Table 4), which could imply a higher conditional probability within the NB classifier if the average tweet length of the three classes are similar, since:

$$\begin{aligned} P(\text{term1}|\text{class1}) &= \frac{(\text{mean freq}) * (\#\text{ instances in class1})}{\text{total }\#\text{ words in class1}} \\ &= \frac{\text{mean freq}}{\text{average tweet length of class1}} \end{aligned}$$

For the examples in the examples in the second category, a possible cause of the misclassification is that those tweets contain the terms "black", 'white", or both, where "LeftTroll" has a much higher mean frequency than the other two classes. (Table 4) Similar to the reason above, these two terms may mislead the NB classifier when classifying these particular tweets.

Term	LeftTroll	RightTroll	OtherTroll
black	0.0502	0.0125	0.0047
white	0.0381	0.0193	0.0051
police	0.0191	0.0165	0.0431

Table 4: "Mean" frequency of the terms in NB model

3 Improving features for NB

3.1 Removing "tweet-id" and "user-id"

According to Table 5 and Figure 2 below, it seems that"tweet-id" has almost no impact on the performance of the NB model, and the proportion of the instances of each class seems to be evenly distributed over the ranges of "tweet-id". For the "user-id" feature, as all of the tweets from a single user appear in the same dataset, even if it is converted into a nominal feature, it could not provide any information about the user in the development or test set when making predictions. Therefore, both "tweet-id" and "user-id" should be removed.

Attribute	Accuracy(%)
With "tweet-id"	60.1167
Without "tweet-id"	60.1274

Table 5: NB model with & without "tweet-id"

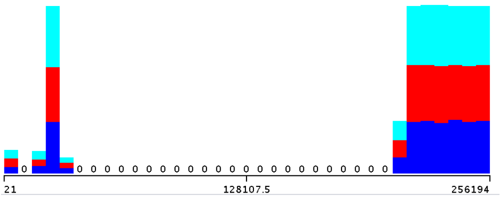


Figure 2: Proportion of the instances of three classes over "tweet-id" in development set (blue: LeftTroll, red: RightTroll, cyan: OtherTroll)

3.2 Numeric to binary

After removing the two id attributes, the NB model could be further improved by converting the numeric attributes (term frequencies as values) into binary attributes, which only indicates the presence or absence of a term in each tweet.

Attribute type	Accuracy(%)
Numeric	60.1274
Binary	65.4767

Table 6: NB models with Numeric and Binary attributes

By converting the attribute type, the accuracy of the NB classifier has increased by more about 5.3%, which strongly supports that the term frequencies are not important for the NB classifier. It is better for the NB classifier to focus on whether the term occurs in the tweet, instead of its frequency, which means that, when the classifier is making predictions, the conditional probability for each class should not be affected by the term frequencies.

4 Conclusion

In conclusion, some knowledge about the two knowledge problems has been gained. NB might be useful for classifying trolls as compared with the baseline and benchmark algorithms used above, and it is most effective on classifying the "OtherTroll" class (with recall 0.971). The NB classifier does not need features "tweet-id" and "user-id", and it is more suitable for NB to use binary attributes instead of numeric attributes (term frequencies).

References

[1] J. E. Cuesta, "Detecting russian troll tweets from the 2016 us,"

[2] C. W. Seah, H. L. Chieu, K. M. A. Chai, L.-N. Teow, and L. W. Yeong, "Troll detection by domain-adapting sentiment analysis," in *Information Fusion (Fusion), 2015 18th International Conference on*, IEEE, 2015, pp. 792–799.

[3] B. C. Boatwright, D. L. Linvill, and P. L. Warren, "Troll factories: The internet research agency and state-sponsored agenda building,"

[4] G. H. John and P. Langley, "Estimating continuous distributions in bayesian classifiers," in *Proceedings of the Eleventh conference on Uncertainty in artificial intelligence*, Morgan Kaufmann Publishers Inc., 1995, pp. 338–345.