Clickbait Detection with Associated Images

Alex Hong
New York University
hong@nyu.edu

Andrew Xiao
New York University
ax296@nyu.edu

Abstract

In recent years, clickbait posts that attract more clicks with sensational posts and lack of quality content have flooded the Internet, distressed users, and had an adverse impact on the reputation of the platform on which they are distributed. In this work, we present a model processing both text and images extracted from an object detection module in order to classify Twitter data. Faster R-CNN with Inception-ResNet v2 is used for the object detection module. We compared the model with a baseline text-only model and investigate the performance among different machine learning algorithms. Inclusion of the images do not lead to improvements in performance in terms of accuracy. We also investigate precision, recall, and F_1 scores among the different methods for the classifier. The classifier can be selected based on the more highly weighted metric.

1. Introduction

As most of the internet business model is based on user-traffic and view count, content producers strive to entice more viewers. On top of creating interesting content, composing attractive headlines is more important than ever before. However, some of the posts go beyond merely having attractive headlines. They tend to have exaggerating or falsifying headlines that mislead the viewers and results in misinformation due to dissonance between the headline and the main article.

These sensational posts are commonly called *Clickbait*. It is a form of yellow journalism focusing on attracting more clicks to the detriment of quality content. The prevalence of clickbait not only distresses users, but also harms the reputation of the platform website on which the clickbaits are posted. In today's digital age, as users are constantly bombarded with information, it is imperative to save users' time by identifying clickbait. In this paper, we propose a text and image based clickbait detecting model and present a performance analysis of different learning methods.

2. Previous Work

Recently, there has been a number of research attempts to detect phishing posts including fake news [21, 4]. The 'Clickbait Challenge 2017[1]' hosted by Bauhaus-Universität Weimar has provided labeled and unlabeled datasets. The Clickbait Challenge resulted in a number of submissions in which research groups attempted to design and implement models through various techniques. For example, Manjesh *et al*. [13] examined the multilayer perceptron (MLP) and neural network with long short-term memory (LSTM) in their research. More recently, the metadata of YouTube videos were used along with variational autoencoders in a semi-supervised way for classification by the Zannettou group [22]. Cao *et al*. [3] identified the length of words in the post text as one of the most important features while Grigorev suggested in his paper the best performing individual model was leveraged on post text[7]. Previous works suggests that post text on its own could serve as a good predictor. While most of the research focused only on text, this study investigated whether the associated images can add more predictive value.

3. Image-Text Combined Classifier

3.1. The Dataset

The data was provided by the organizers of Bauhaus-Universität Weimar for *Clickbait Challenge*. The data consists of Twitter tweets, timestamp, summarizing keywords, and associated images. While there were two labeled and one unlabeled datasets, only the labeled dataset was examined. The two labeled datasets consist of 2,495 posts and 19,538 posts. For this paper, only posts containing images were extracted, which totaled up to 11,429.

There was a continuous as well as a categorical (binary) label. Each post was manually examined by five annotators who scored each tweet based on a 4 level score [0, 0.3, 0.66, 1]. The designation of the class is not related to the scaled average. In this experiment, we focused on the binary class label.

[1] https://www.clickbait-challenge.org/

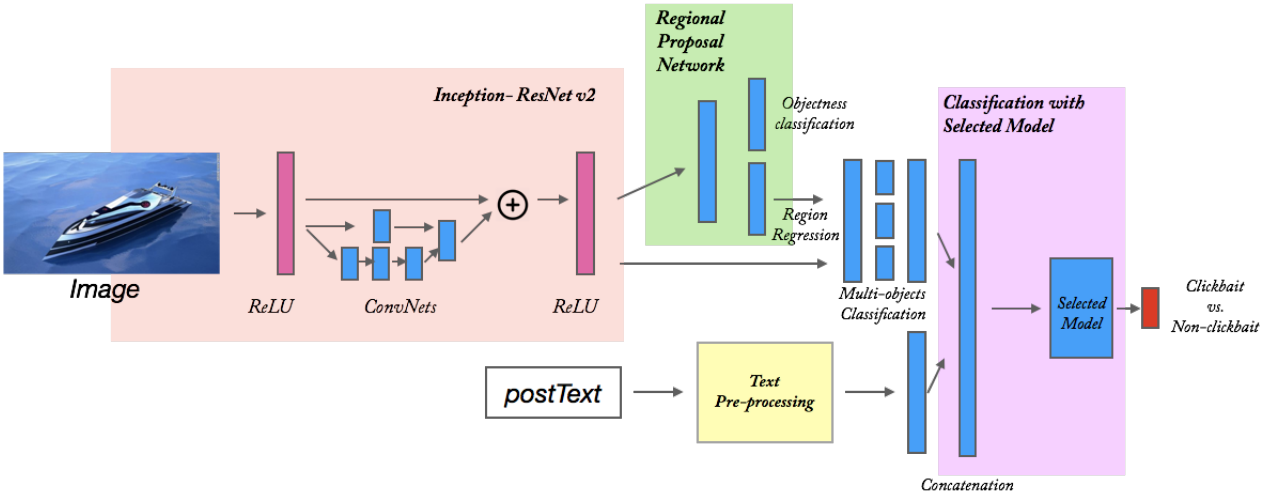

Figure 1. Preparing the image and text for classifier. Image passes through the deep neural network of Inception-ResNet-v2, followed by feature extraction in Regional Proposal Network and Multi-object classifier. The text is pre-processed and vectorized by bag-of-words. The output from image and text processing are concatenated and fed into the classifier.

3.2. The Model Architecture

The architecture can be described in three parts: extracting the features from image, getting text input after vectorization, and building classifier incorporating image and text representation coming from aforementioned components (Figure 1). For final classifiers, we experimented with models based on algorithms that have been commonly used and state of the art, such as Decision Tree, Naive Bayes, Logistic Regression, Gradient Boosting, Random Forest, and Neural Network [2, 6, 1].

Decision tree is a nonlinear classifier, and is efficient computationally. In splitting each tree, it is imperative to maximize the information gain and minimize the entropy. In order to prevent overfitting, we use an ensemble of trees in a random forest, where each tree learns using a random subset with replacement of the training data. The split occurs at a random subset of the features. On average, the variance is reduced in the model. The final prediction is dependent on the average of the tree classifiers. Unlike the random forest, in which each tree is trained on its own, gradient boosting considers the sequence of the weak learning trees[6]. At each step, each model is trained based on the error observed previously.

Naive Bayes is a linear classifier that makes a different assumption in that the features are conditionally independent of the classes. In contrast to logistic regression where parameters of $P(Y|X)$ can be estimated, it is a generative model that can generate data sample and uses Bayes' rule to compute $P(Y|X)$. Naive Bayes can be characterized as follows

$$\hat{y} = \underset{y}{\operatorname{argmax}}(\log \hat{p}(Y = y) + \sum_{j=1}^{m} \log \hat{p}(X_j = x_j | Y = y)$$

where m is the number of features, and each x_j is a vector of inputs and y is the binary label.

Logistic regression is commonly used to predict among two or more classes. As a consequence of the sigmoidal properties of the function, we cannot predict using the mean squared error. Instead, the cross entropy function is employed and consists of two monotonic functions which facilitate the computation of gradient. The logistic cost function predicts a probability value between 0 and 1 and as the predicted value diverges from the truth label, the log loss increases. This can best be represented with the following expression where m is the number of samples, h representing the sigmoid function, and θ a vector of weights

$$\mathcal{L}(\theta) = \frac{1}{m} \sum_{i=1}^{m} \left(y^{(i)} \log(h_\theta(x^{(i)}) + (1 - y^{(i)}) \log(1 - h_\theta(x^{(i)})) \right)$$

3.3. Image Representation

The associated images contain various objects and each object may appear multiple times and are unlabeled. If we attempt to classify those images, it will be intractable because different combinations of the number of object appearances will result in different classes. Instead, applying object detection is appropriate for this task. In other words, we extracted a tensor storing the frequency of occurrence of each object that will be used for feeding into the final part of the model.

Inspired by success in image classification, deep neural networks are also widely used in object detection [11]. Among deep neural network based approaches, we have used Faster R-CNN [16]. R-CNN [7] extracts regions of interest through selective search [20] and forwards the region through convolutional layers. The model has shown good performance but its weakness stems from runtime limitations. Faster R-CNN overcomes this issue by generating regional proposals using neural networks called *region proposal networks* (RPN) [16]. The model is more powerful than other popular models like SSD[12] or YOLO[15], in terms of mean average precision (mAP) [10]. The RPN takes an image of arbitrary size and outputs an objectiveness score for each region. For each anchor a, which is a collection of regions (or boxes), if we find groundtruth region b then set class label of a as $y_a \in \{1...K\}$ and assign a region encoding b with respect to anchor a ($\phi(b_a; a)$)[10]. If b is not found, then set y_a to be zero. Thus, if we predict region encoding $f_{reg}(\mathcal{I}; a, \theta)$ and corresponding class $f_{cls}(\mathcal{I}; a, \theta)$, where $\mathcal{I}$ is the image and θ model parameters, then the loss function is denoted by

$$\begin{aligned}\mathcal{L}(a, \mathcal{I}; \theta)) =& \mathbf{1} \cdot \ell_{reg}(\phi(b_a; a) - f_{reg}(\mathcal{I}; a, \theta)) \\ &+ \lambda \cdot \ell_{cls}(y_a, f_{cls}(\mathcal{I}; a, \theta))\end{aligned}$$

where λ is weight-balancing parameter. At the multi-object classification phase, the loss function is reused only with regional proposals generated from RPN in order to predict a class and its regional refinement.

For convolutional layer prior to RPN, the original paper adopts the model developed by Zeiler and Fergus [23] using VGG-16 [17]. However, for better performance in terms of accuracy and speed, we implemented Inception-ResNet v2 [18, 10]. The architecture is a hybrid of ResNet [9] and Inception [19] as it keeps residual connections and constructs wider networks.

3.4. Text Representation

As mentioned earlier, most of the previous work on this dataset explored text representation. It has been observed from the Clickbait challenge results that applying state-of-the-art architecture such as recurrent neural network did not show distinctive difference compared to other models relying on conventional techniques. Intuitively, it can be inferred that the crudeness of Twitter post, prevalence of grammatically false expressions, arbitrarily abbreviated expressions, and emojis might have an adverse impact on training the context by neural networks.

The pre-processing stage involved tokenizing each post, converting upper case words to lower case, removing delimiters, punctuations, HTML markup, and emojis. Although the default list of stop words from the *Natural Language Toolkit*[2] was considered to further reduce the number of features, the concern was that certain words important to classifying clickbait would be removed. To resolve this concern, word stemming was applied, resulting in a reduction in the size of the vocabulary. Using the bag of words model as first coined by Harris in the 1950s [8], the resulting text was vectorized to convert text into a numerical representation. The bag of words model does not concern itself with the sequence of words in text and can be mathematically summarized as follows

$$P(Y) = \underset{y}{\operatorname{argmax}}(P(y) \prod_{i=1}^{n} P(x_i|y))$$

where n is the number of documents and x_i is a vector of words in the i^{th} document. The following step concatenated this data containing the words as features with the features extracted from the image identification process for model construction.

[2] https://www.nltk.org/

Model	Text	Text&Image
Decision Tree	0.7730	0.7682
Naive Bayes	0.7957	0.7944
Logistic Regression	0.8237	0.8259
Gradient Boosting	0.8106	0.8123
Random Forest	0.8066	0.8053

Table 1. Classification accuracy on text-only dataset and text-and-image dataset.

4. Experiments

4.1. Evaluation

The dataset was randomly split into training and testing subsets in a 4:1 ratio. The training set was split again for 10-fold cross validation. The experimental goal was 1) to see whether image can enhance the accuracy and 2) which algorithm achieves the best performance. The baseline model only considers the text representation and excludes the object detection component. The object detection module for extracting the features was pre-trained on the Common Object in Context (COCO) dataset[3] which contains 80 classes. We used a confusion matrix for measuring the performance in cross validation on the training set. Because the datasets containing less 'clickbait' posts, identifying accuracy does not adequately capture the performance [14]. For binary classification task, Receiver Operator Characteristic (ROC) curves are commonly used, but when dealing with highly skewed datasets, precision and recall gives better expressiveness in terms of performance [5]. Thus, we focused on precision and recall when comparing the classifier models.

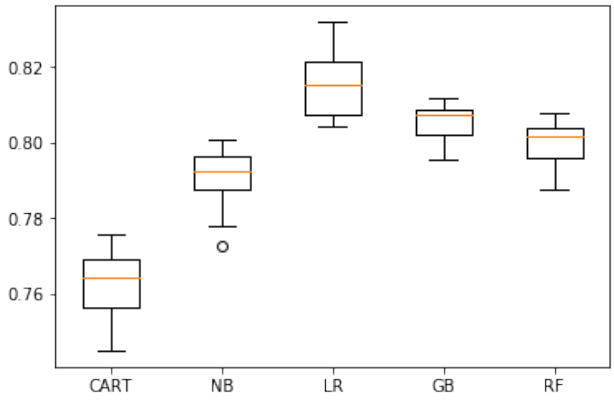

Figure 2. Model comparison on text in terms of accuracy

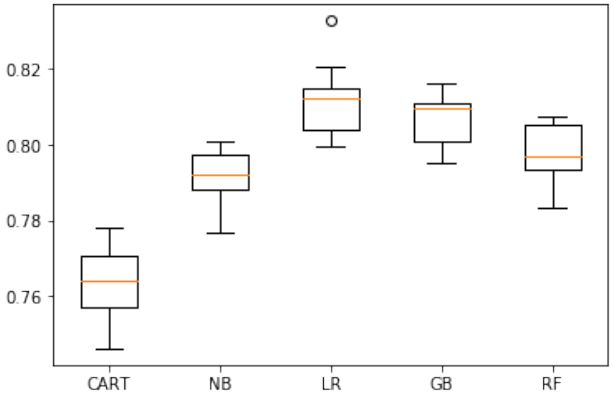

Figure 3. Model comparison on text and images in terms of accuracy

4.2. Performance Analysis

In this study, the images do not add to the performance of our model. As illustrated in Table 1, there seems to be negligible differences in accuracy when comparing text and images to the text only model. In addition, a neural network with two convolutional layers and one batch normalization was also tested with no improvement. The results from cross validation are not much different (Figure 2, 3). This result can be inferred as the image could not give additional information gain for classification task. In fact, associated images share many features with the corresponding texts and in concatenated tensor 38 features of 80 features from image share with those from the text. In other words, if a certain word appears in the post, it is likely for the associated image to contain the object of word, regardless of the number of occurrences.

Among tested models, logistic regression has the best accuracy as illustrated in Figure 3. However, its precision and recall could not outperform gradient boosting and Naive Bayes respectively (Table 2). If the objective focuses on maximizing the number of true clickbait among posts classified as clickbait, i.e. minimizing Type I error, gradient boosting would be the appropriate classifer. On the other hand, if the objective is to maximize the number of clickbait, i.e. minimizing Type II error, Naive Bayes would be the best. However, both gradient boosting and Naive Bayes have limitations. Although gradient boosting has the best precision, its recall is the lowest. It implies that even though it is highly likely to be clickbait if the model classifies as clickbait, the model misses the majority of authentic clickbait. While Naive Bayes has the highest recall among models, the number cannot be seen as distinctively better than other models.

The F_1 score is introduced as it considers the harmonic mean of precision and recall. Logistic regression has shown the best performance in this metric because it has balanced performance in both precision and recall and can be considered to be the more robust model. Gradient boosting has the lowest value in this metric due to significantly low recall with high precision. This implies that unless the detection focuses on having high precision, the gradient boosting method should be avoided.

Model	Precision	Recall	F_1
Decision Tree	0.48	0.43	0.45
Naive Bayes	0.55	**0.51**	0.53
Logistic Regression	0.65	0.48	**0.55**
Gradient Boosting	**0.73**	0.26	0.39
Random Forest	0.65	0.29	0.40

Table 2. Precision, recall, and F_1 analysis of classification on text-and-image dataset.

5. Conclusion and Future Work

In this paper we have shown that there does not appear to be any significant gain in accuracy from images in improving our model for detecting clickbait. Logistic regression had a slight, but mostly negligible edge over the other methods. It is sensible that the images associated with text would share similar features, many of which would be repetitive.

We also have shown that simply using accuracy in binary classification task can be misleading. By using precision and recall evaluation metrics, a more informed and appropriate model can be selected.

For future work, it would be interesting to work with bigrams and trigrams as the sequence of text conveys meaning. Furthermore, while emojis were not included during the text processing phase, the sheer diversity of emojis today contain meaning that may convey pivotal information. It is also notable that the image training was performed on a limited set of images spanning 80 classes, which may

[3] http://cocodataset.org/

not capture the nuances that uniquely characterize clickbait. An alternative would be to explore other potential image datasets for pretraining purposes.

References

[1] L. Breiman. Random forests. *Machine learning*, 45(1):5–32, 2001. 2

[2] L. Breiman. *Classification and regression trees*. Routledge, 2017. 2

[3] X. Cao, T. Le, and J. Zhang. Machine learning based detection of clickbait posts in social media. *CoRR*, abs/1710.01977, 2017. 1

[4] N. J. Conroy, V. L. Rubin, and Y. Chen. Automatic deception detection: Methods for finding fake news. *Proceedings of the Association for Information Science and Technology*, 52(1):1–4, 2015. 1

[5] J. Davis and M. Goadrich. The relationship between precision-recall and roc curves. In *Proceedings of the 23rd international conference on Machine learning*, pages 233–240. ACM, 2006. 3

[6] J. H. Friedman. Greedy function approximation: a gradient boosting machine. *Annals of statistics*, pages 1189–1232, 2001. 2

[7] A. Grigorev. Identifying clickbait posts on social media with an ensemble of linear models. *CoRR*, abs/1710.00399, 2017. 1, 3

[8] Z. Harris. Distributional structure. *Word*, 10(23):146–162, 1954. 3

[9] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 3

[10] J. Huang, V. Rathod, C. Sun, M. Zhu, A. Korattikara, A. Fathi, I. Fischer, Z. Wojna, Y. Song, S. Guadarrama, et al. Speed/accuracy trade-offs for modern convolutional object detectors. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7310–7311, 2017. 3

[11] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012. 3

[12] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg. Ssd: Single shot multibox detector. In *European conference on computer vision*, pages 21–37. Springer, 2016. 3

[13] S. Manjesh, T. Kanakagiri, P. Vaishak, V. Chettiar, and G. Shobha. Clickbait pattern detection and classification of news headlines using natural language processing. In *2017 2nd International Conference on Computational Systems and Information Technology for Sustainable Solution (CSITSS)*, pages 1–5. IEEE, 2017. 1

[14] F. J. Provost, T. Fawcett, R. Kohavi, et al. The case against accuracy estimation for comparing induction algorithms. In *ICML*, volume 98, pages 445–453, 1998. 3

[15] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi. You only look once: Unified, real-time object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 779–788, 2016. 3

[16] S. Ren, K. He, R. Girshick, and J. Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in neural information processing systems*, pages 91–99, 2015. 3

[17] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014. 3

[18] C. Szegedy, S. Ioffe, V. Vanhoucke, and A. A. Alemi. Inception-v4, inception-resnet and the impact of residual connections on learning. In *Thirty-First AAAI Conference on Artificial Intelligence*, 2017. 3

[19] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna. Rethinking the inception architecture for computer vision. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016. 3

[20] J. R. Uijlings, K. E. Van De Sande, T. Gevers, and A. W. Smeulders. Selective search for object recognition. *International journal of computer vision*, 104(2):154–171, 2013. 3

[21] S. Volkova, K. Shaffer, J. Y. Jang, and N. Hodas. Separating facts from fiction: Linguistic models to classify suspicious and trusted news posts on twitter. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 647–653, 2017. 1

[22] S. Zannettou, S. Chatzis, K. Papadamou, and M. Sirivianos. The good, the bad and the bait: Detecting and characterizing clickbait on youtube. In *2018 IEEE Security and Privacy Workshops (SPW)*, pages 63–69. IEEE, 2018. 1

[23] M. D. Zeiler and R. Fergus. Visualizing and understanding convolutional networks. In *European conference on computer vision*, pages 818–833. Springer, 2014. 3