Creating a machine learning algorithm to detect clickbait articles

Joshua Zbozny
Mentored by Mr. Gregory Gray

Introduction

With the increase in prevalence of online media and news, immediately capturing attention is crucial. Online news outlets are paid based on advertisements that are seen as a result of a click on their website, so the competition for clicks is incredibly high. This has led to the rise of misleading titles called clickbait that exist for the all-important click. An article is clickbait if its title was made to be sensationalized or purely made to entice the reader into clicking on a link. A well-crafted clickbait title will tell a consumer just enough information to intrigue them, while giving them nothing more, forcing the reader to click on the link and read the article if they want to learn more. This is known as the Curiosity Gap and clickbait articles exploit this. This can be harmful to a consumer as it causes "cognitive overload, deterring the readers from reading more informative and in-depth news stories" (Chakraborty, Paranjape, Kakarla, & Ganguly, 2016).

The purpose of this project was to create a machine learning algorithm that can detect if a given article is clickbait. The algorithm needed to differentiate with an accuracy of at least 80% to be successful.

Methods and Materials

Following the main methods of machine learning, the data was pre-processed and then the algorithm was created, improved, and tested. The data was obtained from a website called Clickbait-Challenge, which held over 19,000 articles that were predetermined to be clickbait or not. The prelabeled data allowed the algorithm to train itself on a portion (2/3) of the dataset and learn what datapoints it predicted correctly and incorrectly. This is how the algorithm was created and refined.

Under the step of data pre-processing, titles were isolated from the rest of the articles, and then stemmed and lemmatized, meaning that only the root of the word was considered for the next steps. Then, a bag of words was created leveraging the new dataset.

A bag of words is a way for a computer to process language (figure 1). It makes an array for each data point in the data set. Each array is made of the occurrences of the top n words in the data set, plus other numeric values that could be used based on the dataset. For this algorithm, the arrays consisted of a count of every occurrence of the top 10,000 words in the dataset and the length, in characters, of the titles.

The dataset was balanced for the minority class clickbait to improve the algorithm. A balanced dataset has the same amount of

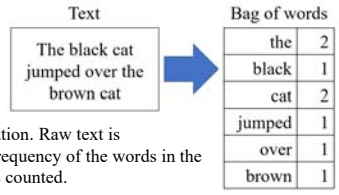


Figure 1 (right): A bag of words representation. Raw text is transformed into an array that counts the frequency of the words in the text. The number of times a word occurs is counted.

Methods and Materials (cont.)

datapoints in each class. With imbalanced data, the algorithm has less examples to learn from in the minority class, making it inherently worse at detecting the minority class. Balancing the data fixes that problem. The algorithm was created in Python, using the scikit-learn library. A Multinomial Naïve Bayes algorithm was selected based on its quickness and accuracy. The Naïve Bayes classifier calculates the probability that each individual feature is clickbait or not. It then chooses the class with the greatest total probability after all the individual features are analyzed.

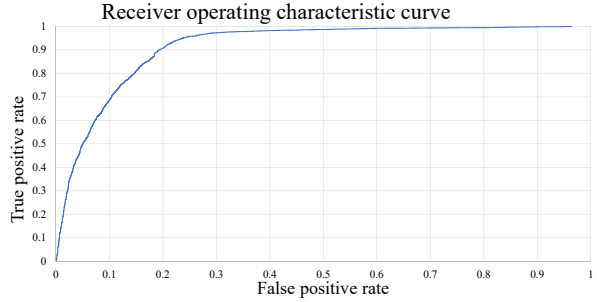
Once the algorithm was created, it was optimized using a RandomizedSearch function, which optimized the internal features of the algorithm through 10-fold cross validation and repeated trials.

An application based on the algorithm was then created using Python. A user can input a news website, and the application will go to the website, and will highlight any title that it determines to be clickbait.

Results

The accuracy was 85% at the end of testing, exceeding the identified goal of 80%. These results can be shown using a table of values accuracy, precision, recall, and f-measure (table 1). Precision is a measure of accuracy. For clickbait, it is the number of articles identified as clickbait that were clickbait over the total number of articles identified as clickbait. Recall is a measure of volume. For clickbait, it is the number of clickbait articles identified as clickbait over the total number of clickbait articles in the dataset. F-measure is a balance between precision and recall. It gives a single statistic to use instead of two.

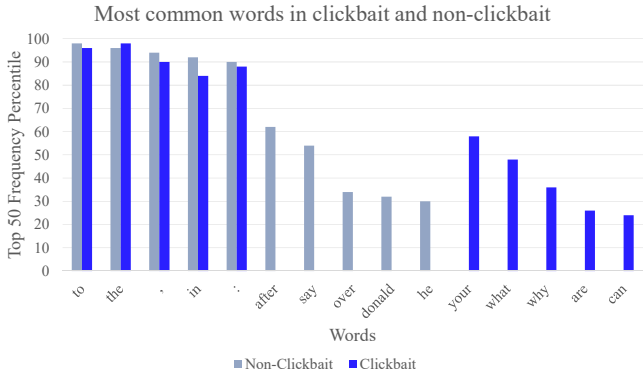
A receiver operating characteristic curve (graph 1) is also shown. The area under this curve, AUC, gives the probability that an article identified as clickbait was a clickbait article.



Graph 1 (above): A receiver operating characteristic curve. This is a display of the true positive rate (a clickbait article successfully identified as clickbait) over the false positive rate (a non-clickbait article identified as clickbait).

Results (cont.)

A graph of the most commonly appearing words in all articles that compares clickbait with non-clickbait as well is shown below (graph 2). The higher up a bar is, the more commonly that word is used. From this graph, it can be seen that very common words are used regardless of context. These words include "the," "to," and "in." Words that give context, like "after" and "say," are used mostly in non-clickbait articles. Finally, words that are more personal, or pique interest are used more commonly in clickbait. These are words such as "your," and "why."



Graph 2 (above): A graph that shows some of the most common words in clickbait or non-clickbait articles, as well as the most common words shared between them.

	Precision	Recall	F-Measure	Accuracy
Clickbait	0.92	0.74	0.82	0.85
Non-clickbait	0.78	0.92	0.85	0.85

Table 1 (above): A display of precision, recall, f-measure, and number of articles for both clickbait and non-clickbait datasets.

Conclusions

The purpose of this project was to create an algorithm that could determine if an article was clickbait or non-clickbait with an accuracy of at least 80%. This was done successfully, showing that it is possible to create and train an accurate machine learning algorithm capable of detecting clickbait articles. An application that was able to search for clickbait articles on a given website and mark them was also created.

References

Chakraborty, A., Paranjape, B., Kakarla, S., & Ganguly, N. (2016). Stop clickbait: Detecting and preventing clickbaits in online news media. *2016 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM)*, 9-16. doi: 10.1109/asonam.2016.7752207