NLP Final Project - Clickbait and Sentiment Analysis

Nahin Mahmood
Hunter College
New York, U.S.A
Nahin.Mahmood27@myhunter.cuny.edu

Tabshir Ahmed
Hunter College
New York, U.S.A
Tabshir.Ahmed70@myhunter.cuny.edu

Rezwan Rahman
Hunter College
New York, U.S.A
Rezwan.rahman05@myhunter.cuny.edu

Abstract

This project addresses a common issue faced by sports fans, particularly soccer enthusiasts, which involves identifying sports news websites that employ the most clickbait headlines and utilize extreme bias or sentiment in their articles. The objective is to save sports fans from wasting their time on misleading information related to their favorite players, teams, or leagues. To achieve this goal, we employed various techniques. Firstly, we implemented a Multinomial Naive Bayes model trained on a CSV dataset containing headlines labeled as clickbait or non-clickbait. This model is able to predict whether a given headline is clickbait. Additionally, we developed models to generate polarity scores for articles, such as its positive, negative, or neutral sentiment values. Furthermore, we created a model that explores the prevalence of hyperbolic words in the articles. Through these approaches, we hope to provide sports fans with insights into which sports news websites are reliable and trustworthy.

I. Introduction

Sport is an acitvity and hobby that has has garnered millions of fans across the world throughout decades. From watching league games to watching major tournaments, fans have continuously been dedicated their times to cheering and following their teams in numerous ways. One of these ways is by spending hours reading news and rumors about their favorite team news through different online media outlets such as SkySports, ESPN, Bleacher Reports, and other domains. This is especially the case through certain time of the season, particularly when the transfer window opens where teams are now able to sign new players or sell their players. Knowing how much sports news can impact the emotions of humans, some media outlets tries to take advantage of this by writing soccer articles and using headlines that are more biased and opinionated rather than fact based. Media outlets steers their article in this direction so they can attract viewers attention to their company and get people to read their article. While it may be beneficial to media outlets, biased soccer articles is a big dilemma for sports fan and soccer clubs. Sports fans are being fed false information that would alters their emotions. For sports clubs, biased articles about their team can create a false perception to the general public, hurting their revenue. Therefore, we propose using our knowledge and principles of Natural Language Process to create models that would be able to categorize soccer articles based on it being clickbait or not clickbait. Our hypothesis is that machine learning models can generally identify sports articles based on their level of clickbait and sentiment.This way, fans of soccer can spend more time being informed about the latest developments in soccer and clubs being correctly represented to the public.

II. Methodology

Along with our Naive Bayes Model, we wanted to have an additional approach to further determine whether or not a sports article is clickbait or not. We then came up with the idea of using Sentiment Analysis for our project. Sentiment Analysis is an NLP approach that gives the different emotional tones being used in any text(positive or negative emotions). We created two rule based Sentiment Analysis Model, Polarity Score Model and Hyperbole Score Model. As our Naive Bayes Model was headline focused, our two Sentiment Analysis Model were article content focused. The first Sentiment Analysis Model we created was a Polarity Score Model. In this model, we would retrieve the content of an article, normalize the text accordingly, and then retrieve the polarity score of each text. Polarity score returns a score of the percentage of positive, negative, and neutral sentiment that is in a text. The second Sentiment Analysis Model we have is a Hyperbole Word Model. In any text, hyperbole words are often used to exaggerate a personal opinion or bias. This should not be the case for any sports articles since these articles are meant to be fact driven articles and not opinionated based. Therefore, any sports article that contains some hyperbole words can surely be considered as a clickbait article. For this model, we retrieved each sports article content and normalized it accordingly. We then would check the number of hyperbole words are present in each article content and divide it by the total amount of words in a text to see the amount of hyperbole word there are in a text. Each of our 3 models outputs a final result based on each article and each media outlet. Each of the model approach will be talked about in details from sections 5-7.

III. Previous Works

Prior to beginning our project, we searched the web to find projects with topics that were similar to ours. Before finalizing our topic as clickbait and sentiment analysis, we

were primarily focused on sentiment analysis as it pertains to sports articles. We were able to find three different well done projects that focus on the sentiment of commentators, and sports articles. Each of these projects utilized natural language processing in order to test and see if their hypothesis was correct. All of these projects will be summarized below.

Reference 1: Investigating Sports Commentator Bias within a Large Corpus of American Football Broadcasts

In this reference, six students from the University of Massachusetts Amherst, and Ursinus College do an investigation on sports commentary bias on American football. They look into players ' race and the commentators' speech to see if there is any racial bias in American football commentary. The students collect football dataset, process broadcast transcripts, identify players' race, naming patterns and explore sentiment patterns among the commentators. This study concluded that white players receive more positive coverage than black players. However Black players are more praised for physical attributes rather than cognitive attributes. There were limitations such as words that gave us unsure understanding of sentiment. What our project would add to the existing body of work, is it that our project will highlight clickbait levels, and place emphasis on exaggerations.

Reference 2: Combining sentiment analysis classifiers to explore multilingual news articles covering London 2012 and Rio 2016 Olympics

In this reference, three people from the International Journal of Digital Humanities do a study where they use multi-lingual corpuses to examine its challenges of working with sentiment analysis. The corpuses are articles covering the 2012 and 2016 olympics. The methodology used to go about this study was three steps: data collection where they collect articles in multiple languages, data preparation where they clean their data in order to remove any limitations that will get in the way of analysis, and analytical methods using sentiment detection techniques. The article concluded that the English corpus was more skeptical about the legacy of the game, whereas non-English corpuses such as Brazilian media was more optimistic. There are also differences when dealing with different corpuses. Text outlets were seen to present more neutral sentiment, and news headlines presented more positive or negative labels. Weakness: there were limitations such as translation error when translating different languages into english. Our project will help to distinguish between overly exaggerated text headlines and non-sensationalized headlines.

Reference 3: Sentiment Analysis of RSS Feeds on Sports; News – A Case Study URL:

In this reference three students from the University of Engineering and Technology in Pakistan do a case study of sentiment analysis on sports news. Particularly RSS feeds. They conduct this study by collecting a corpus of 5 different sports groups, look at emotion values in text and subjectivity to analyze the sentiment. In conclusion they found that hockey had the least positive polarity, cricket had the highest negative polarity, football had the least negative polarity, and American football had the highest positive polarity. Weakness: dealing with a wide assortment of texts, a lot of unstructured text raised the need for data cleaning, making it more difficult to go about processing. Our project will focus on one particular sport, allowing us to deal with less limitations.

Through these different projects, we got a good idea on how we can go about working on our project. These different projects tackle unique results as it pertains to sentiment analysis. One focuses on racial sentiment in sports, the other focuses on comparing different countries' sentiment towards sports, and the last one focuses on a case study to see the positive and negative polarity of five different sports. We decided to focus on clickbait titles, and sentiment analysis of those clickbait articles to see what articles are clickbait and which aren't.

IV. Data Collection

For our Naive Bayes Model, we will be using two different datasets. To train this model, we obtained a dataset from Kaggle.com that contains 32,000 headlines from different news article as shown in Figure 1. In this dataset, the headlines are broken evenly into two categories; 16000 article headlines that are classified as "clickbait" are given the number 1 and 16000 article headlines that are classified as "non clickbait" are given the number 0.

Fig. 1. Results of Hyperbole Word

Which TV Female Friend Group Do You Belong In	1
The New "Star Wars: The Force Awakens" Trailer Is Here To Give You Chills	1
This Vine Of New York On "Celebrity Big Brother" Is Fucking Perfect	1
A Couple Did A Stunning Photo Shoot With Their Baby After Learning She Had A	1
How To Flirt With Queer Girls Without Making A Total Fool Of Yourself	1
32 Cute Things To Distract From Your Awkward Thanksgiving	1
If Disney Princesses Were From Florida	1
What's A Quote Or Lyric That Best Describes Your Depression	1
Natalie Dormer And Sam Claflin Play A Game To See How They'd Actually Last In	1
16 Perfect Responses To The Indian Patriarchy	1
21 Times I Died During The "Captain America: Civil War" Teaser	1
17 Times Kourtney Kardashian Shut Down Her Own Family	1
Does Coffee Make You Poop	1
Who Is Your Celebrity Ex Based On Your Zodiac	1
17 Hairdresser Struggles Every Black Girl Knows To Be True	1
Are You More Walter White Or Heisenberg	1
The Most Canadian Groom Ever Left His Wedding To Plow Out His Guests In A S	1
Here's One Really Weird Thing About Butterfree	1
15 Resolutions To Make Good On In 2016	1

For testing the Naive Bayes Model, we used 40 sports articles URL as shown in Figure 2. We obtained 10 articles each from the following medias; ESPN.com, SkySports.com, CBSSports.com, and Control Websites. Control websites included 10 soccer articles from different media outlets. For the purpose of the training model (Naive Bayes Model) , we scraped the 40 sports article so that we can retrieve its headline only, thus making our testing data 100 headlines.

For testing our two rule based Sentiment Analysis Model, we used the same 40 articles from ESPN.com, CBSSports.com, SkySports.com, and Control Websites. For the purpose of our Sentiment Analysis models, we scraped each one of the articles manually to obtain the article body, thus making our dataset consist of 40 article contents.

The URLS used in the Sentiment Analysis Models and Naive Bayes Testing Model can be found below

V. Naive Bayes Model

Algorithm

To construct our Multinomial Naive Bayes Model, a text classifier designed to detect the probability of a headline being clickbait, we utilized the Pandas library for text preprocessing on the "headlines" column. We tokenized the headlines, converted them to lowercase, and subsequently eliminated stop words and punctuation. For model training, our primary tool was the Sklearn library, enabling us to predict the likelihood of a headline being clickbait.

To train the model effectively, we partitioned our dataset, allocating 20 percent for testing and utilizing the remaining 80 percent for model training. Employing machine learning pipelining, we first utilized CountVectorizer to transform our dataset into a matrix of token counts. This enabled us to subsequently apply the TfidfTransformer, facilitating an analysis of word importance within each document. By assigning lesser weight to frequently occurring words and greater weight to rare words, we could effectively identify indicators of clickbait headlines.

We employed the Multinomial Naive Bayes algorithm to generate a model capable of predicting the probability of a headline being clickbait. The model was trained using the "fit" method of the pipeline on the training data. To generate predictions, we utilized the "predict" method of the pipeline on the testing data. To evaluate the model's accuracy, we employed the "accuracyscore" function from scikit-learn, computing the accuracy as a percentage. The resulting accuracy was printed to the console, which happened to be surprisingly high 96.19 percent. To get the clickbait probability score for each headline, we used the ".predictproba()" function and attached it along with the text classifier with the input of the function being the headline.All coding pertaining to the model is encapsulated within the file named "clickbaitheadlinetrainingmodel.py".

Using the Model to Generate results

To effectively utilize our model for this study to generate results, we collected 10 URLs from each sports news website, ensuring that each URL number pertained to a similar topic covered by all the sports news websites. To be able to extract the headlines/titles from a URL, we used the Beautiful Soup and HTTPS Requests libraries to extract all the texts in the title tags from the urls. Afterwords, we used the Multinomial Naive Bayes Model to parse through each url in the array that corresponded to each sports news website to get a probability for each article being clickbait and also created an overall average clickbait probability of all the articles headline from a sports news website. We used the overall average clickbait probability to make comparisons and identifying which sports news websites we looked at used the least and most amount of clickbait. One important thing to note is that in case we are unable to extract the title from a website, we ignore it from calculating the overall average clickbait probability for that sports news website.

VI. Sentiment Analysis - Polarity Score

Data Retrieval

In our Polarity Score rule based model, we first retrieved each of the 40 articles content by calling each variable from our Raw data.py file (The figure can be found below). In our Raw data.py we named each variable according to the website where the article was from(ex: ss for SkSports). Each variable in this file contained a specific article content that was scraped manually by us. For the purpose of our Polairty Score model approach, we wanted to see the percent of postive, negative, and neutral sentiment for each article. We also wanted to see the polarity score of each article in relation to other articles from the same media outlet. Therefore, in our datafram we seperated into 4 groups that would attain the polairty score of articles that are only from that specific outlet. . Creating this dataframe of seperating the polarity score of the media outlets will allow us to see which media outlets has the most sentiment and see each media site in a deeper level through each one of its article.

Algorithm

In order for us to get the best possible sentiment score, we must go through a process of text editing. Our initial step for this model was to normalize our data set. To do this, we make our text lowercase, remove punctuation, tokenize our text, remove stopwords, and then detokenize our text. The reason why we detokenized our text at the end of the normilization was because we wanted to get the score of the whole article as one, not getting the score by seeing the individual score of each word used in the article. We then import the librayr sentimentintensity in order to use the function polarityscore(). This function serves as the analyzer to get the polarity score. After getting the polarity score of all articles, we stored all our polarity scores within our dictionary. We did this for every article used in our data collection and built a data frame to visualize all our results:

VII. Sentiment Analysis - Hyperbole Word

Data Retrieval

In our Hyperbole Word rule based model, we first retrieved each of the 40 articles content by calling each variable from our Raw data.py file (The figure can be found below). In our Raw data.py we named each variable according to the website where the article was from(ex: ss for SkSports, espn for ESPN, cbs for CBSsports, and cw for Control Websites). Each variable in this file contained a specific article content that was scraped manually by us. For the purpose of our Hyperbole Model approach, we wanted to see the percentage of Hyperbole words used as an average for each media outlet. Therefore, we separated the 40 URL variables into 4 dictionaries; ss article, espn articles, cbs articles, and cw articles, with each dictionary containing 10 variables containing URLs from their respective media outlet. We also created a dictionary that contained every single article from every media outlet. Creating this dictionary of all URLS would allow us to see the overall bulk of hyperbole words being used in sports article throughout.

Fig. 2. Rawdata.py

```
skysports = {
    "https://www.skysports.com/football/news/11095/12872241/lionel-messi-to-leave-p
    "https://www.skysports.com/football/news/11661/12867418/erling-haaland-man-city
    "https://www.skysports.com/football/udinese-vs-napoli/report/470228",
    "https://www.skysports.com/football/news/11095/12872161/jude-bellingham-real-ma
    "https://www.skysports.com/football/news/11668/12872729/chelsea-transfer-news-a
    "https://www.skysports.com/football/news/11095/12867919/qatari-group-led-by-she
    "https://www.skysports.com/football/news/11675/12873898/julian-nagelsmann-totte
    "https://www.skysports.com/football/news/11661/12804623/man-city-premier-league
    "https://www.skysports.com/watch/video/sports/football/12873722/mikel-arteta-i-
    "https://www.skysports.com/football/news/11095/12865122/barcelona-make-chelseas
}

espn = {
    "https://www.espn690.com/sports/messi-apologizes-psg/7XV2TT7XAM6HNJAKHG3AZI5DAI
    "https://www.espn.com/soccer/manchester-city-engman_city/story/4940481/man-city
    "https://www.espn.com/soccer/blog-marcottis-musings/story/4939861/what-napolis-
    "https://www.espn.com/soccer/soccer-transfers/story/4939553/summer-transfer-pre
    "https://www.espn.com/soccer/chelsea-engchelsea/story/4932909/chelseamauricio-p
    "https://www.espn.com/soccer/blog-transfer-talk/story/4941511/transfer-talk-wes
    "https://www.espn.com/soccer/bayern-munich-gerbayern_munich/story/4908679/bayer
    "https://www.espn.com/soccer/soccer-transfers/story/4870556/man-city-charged-ov
    "https://www.espn.com/soccer/english-premier-league/story/4935721/when-can-manc
    "https://www.espn.com/soccer/soccer-transfers/story/4733490/pierre-emerick-auba
}
```

Data Normalization

To normalize the data sets we retrieved from rawdata.py, we first added a .lowercase() function to each variable containing an article content data. Lowercasing data creates simplicity and a more consistent output for accurate results. Lowercase is also good practice since it can be the determiner of misinterpreting a word. We then created a function regularpunct. In this function, it checks if each variable that contains an article has any punctuation marks. If it does, then it will remove the punctuation mark present in the article content and replace it with a blank space. This is necessary because the main objective is to focus on the words being used in the data. After the words were lowercase and all punctuation marks were removed, we then tokenized every data with the built in function wordtokenize(). This is done so that the data set can now become a list that separates every word instead of combining all words to keep as a single string. The last step of our data normalization was to remove any stopwords that were present in each data by using stopwords.words('english'). This function would then check for any stopwords and add the non stopwords to the original tokenized list. Removing stopwords allows us to focus on the more important information and words for our model as opposed to unimportant words altering our dataset results.

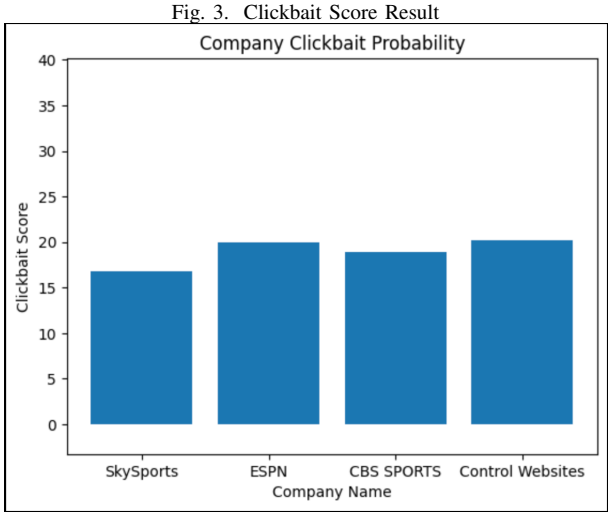
Algorithm

To create our rule base approach, we first created an array called hyperbolewords that contained the most prominent hyperbole words used in English Language, both positive and negative based. After the array was made, we then made a separate rule based function where we would retrieve a score based on hyperbole words present in each article content. To do this, we first call every variable containing an article content(ex ss1) and count the number of words that matches to the words in our hyperbolewords array. To do this, we used the method sum(example.count(x) for x in (hyperbolewords)). This specifically calls a variable x that calls to check and count for every hyperbole word present in the article text and obtain the sum of all of them. After getting the sum of hyperbole words present, we then divide the sum of the hyperbole words present in the article by the total number of words in the article. This gives us a percentage of the amount of hyperbole words present in the article. We multiply this percentage number by 10. The reason why 10 is being multiplied will be touched in Section 7. We finally rounded the number we received in the previous formula and round it to two decimals with the round() function to get the hyperbolescore for that specific article content. After repeating this step for every variable containing an article content, we then stored the 40 Hyperbole scores we got into 4 different arrays. The 4 arrays were separated to hyperbole scores for each article in their respective media outlet; sshyperbole array for all SkySports article hyperbole score, espnhyperbolearray for all ESPN article hyperbole score, cbshyperbolearray for all CBSsports article hyperbole score, and cwhyperbolearray for all Control Websites score. We then averaged these 4 array to get the final hyperbole score for each media outlet and used these 4 scores as our final result for our Hyperbole Rule Based Model.

VIII. Results and Discussion

Naive Bayes Model

Fig. 3. Clickbait Score Result



A. Analyzing the Results

In this project, we focused on analyzing specific prominent sports news websites, namely SkySports, ESPN, CBS Sports, and Control Websites. The Control Websites served as a comparison group, representing other websites where we intentionally tried to pick out the most clickbait titles as a baseline for our evaluation. Our aim was to determine the performance of our model by ensuring that this particular column exhibited the highest average clickbait probability.

According to the data and visualizations in Figure 3, we observed that Control Websites had the highest average clickbait probability at 20.15. ESPN followed closely behind at 19.93, making it the most prominent offender for clickbait usage from three sports news websites we looked at. CBS Sports

was next with a clickbait probability of 18.85, and SkySports had the lowest average clickbait probability at 16.74. We also collected data on the clickbait probability for each individual article from each company. A surprising finding was that one of the SkySports article headlines was identified by the model to have the highest clickbait probability at 86.88, surpassing all other articles the model looked at. This result was unexpected considering that the overall average clickbait probability for SkySports was the lowest among the analyzed websites.

B. Limitations of the Model

While our model has shown promising results, it is important to acknowledge its shortcomings. Firstly, the dataset we used, obtained from Kaggle, comprises headlines from various types of articles, not solely focused on sports. While this diversification can be advantageous, the Multinomial Naive Bayes Model might be better used for non-sports-related articles. As a result, the dataset contains a limited number of sports article headlines, resulting in the Multinomial Naive Bayes Model assigning relatively low clickbait probability scores to sports headlines by default. Additionally, one of the most effective techniques sports news websites employ for clickbait is incorporating false information associated with a prominent player or team. Unfortunately, the model is not designed to determine whether the headline is deliberately spewing misinformation. It is important to mention these potential limitations the model has in context to sports news in general.

Sentiment Analysis - Polarity

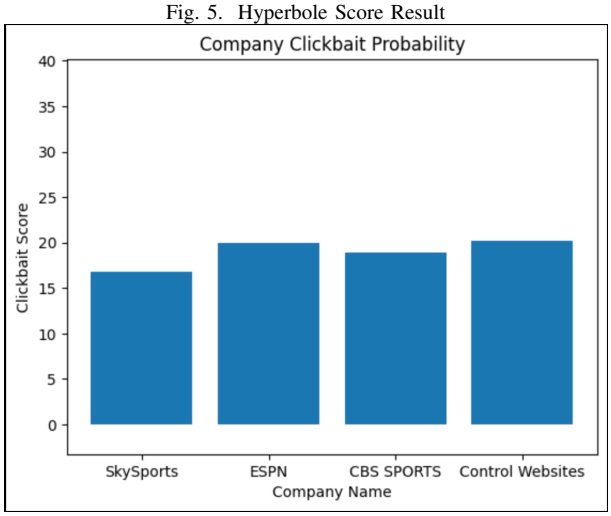
Fig. 4. Polarity Score Result

Polarity Score
{'neg': 0.067, 'neu': 0.751, 'pos': 0.182, 'co...
{'neg': 0.065, 'neu': 0.85, 'pos': 0.085, 'com...
{'neg': 0.018, 'neu': 0.772, 'pos': 0.21, 'com...
{'neg': 0.047, 'neu': 0.77, 'pos': 0.182, 'com...
{'neg': 0.088, 'neu': 0.798, 'pos': 0.114, 'co...
{'neg': 0.083, 'neu': 0.748, 'pos': 0.169, 'co...
{'neg': 0.037, 'neu': 0.788, 'pos': 0.175, 'co...
{'neg': 0.098, 'neu': 0.781, 'pos': 0.121, 'co...
{'neg': 0.0, 'neu': 0.933, 'pos': 0.067, 'comp...
{'neg': 0.004, 'neu': 0.813, 'pos': 0.183, 'co...
{'neg': 0.095, 'neu': 0.757, 'pos': 0.148, 'co...
{'neg': 0.083, 'neu': 0.732, 'pos': 0.185, 'co...

For the sentiment analysis polarity score rule based approach, our final results were based on using all our articles individually, going through our series of steps listed above to remove unwanted information, and use sentiment intensity analyzer to generate a positive, negative, neutral score. As showin in the figure 4, every article had a high neutral polarity score for the most part as this was expected. However, we noticed that articles with lower clickbait score had higher positive polarity scores than negative. Our SkySports article on Mikael Arteta which had the highest click bait score had a 0 negative polarity score. For most of the articles that have a moderate (15-50) clickbait score don't have a huge disparity when it comes to positive to negative clickbait ratio. Most articles had a higher positive polarity. The reason we took this approach to see if our clickbait score of the title has any correlations with the polarity score of the article. More than 95 percent of the articles don't, whereas a few do.

Sentiment Analysis - Hyperbole

Fig. 5. Hyperbole Score Result



For our Sentiment Analysis Hyperbole word rule based approach, our final results were based on averaging all the hyperbole score for each of the media outlet, them being ESPN, CBSsports, SkySports, and Control Websites. Before getting into these 4 results, as previously mentioned we multiplied the result of the sum of all hyperbole words present in each article content divided by the total number of words in the article content. The reason why we took this approach is because we wanted to give a score based on the scale range of 0-1, as this would show users a more precise result. Therefore. by multiplying each result by 10 it gave us scores within this range.

Based on the results shown in Figure 5, we can say that since SkySports articles has the highest likelihood of being a clickbait article based on Hyperbole word model while ESPN has the least likelihood of being a clickbait article based on Hyperbole word model. From our research, this model is accurate in terms of determining whicch media outlets/ articles are potentially clickbait. It is well know that ESPN only releases articles that are fact based, so them being having the least amount of hyperbole words is concise. Skysports having the highest amount of hyperbole words used is also accurate based on our manual research. SkySports articles tend to use some type of hyperbolic words to exaggerate any point the author is trying to make. Although using this approach is a good method/feature, it cannot be the main condition for determining whether a sports article is clickbait or not. Even though its unlikely to occur, some articles may use hyperbole words to not be bias all the time but more so to emphasize the fact they are stating. The author may do this to catch the attention of the readers but at the same time not use hyperbole words in order to persuade them of an opinion, but more so emphasize a fact. Along with this, the Sentiment Analysis

Hyperbole model could of used more data in order to make the results more accurate and credible. Using 40 URL Links is a great start, but it cannot be the total amount of data used. There are millions of sports article surfacing in the internet, as in order for this model to be more reliable, it needs to be fed with more data to cross validate with whether or not the outputs are correct.

IX. Conclusion

In this project, we applied the knowledge learned from this semester to create an approach in determining whether a sports article can be considered as clickbait or not. We created 3 different models that gave us scores of different aspects of each article in determining whether the article can be classified as a clickbait article or not. The ideal vision we had in mind was by using our 3 different models for the same data sets, we would be able to create an overall accurate prognosis of whether an article can be classified as a clickbait or not. Our Naive Bayes Model was first trained given a large data set and resulted in a high accuracy score. Then we were able to use this model to give a prediction of whether an article can be classified as clickbait based on it's headline. From using headlines as our main data sets, we then shifted to using the article text as the data sets for our other two models. For our first Sentiment Analysis model, Polarity Score Model, we were able to create a rule based approach where we would obtain the polarity score of each article that would tell use the percentage of positive, negative, and neutral sentiment were present in the article. For our second Sentiment Analysis model, Hyperbole Word Model, we were able to create a rule based approach that would help us know the amount of exaggeration that are present in each article by obtaining the number of hyperbole words in each articles and dividing that by the total amount of words in each article. Each one of these models ties together to bring an overall understanding of whether or not a specific sports article or a specific sports media outlet can be classified as clickbait or not.

X. Future Work

Unfortunately, due to lack of time and lack of people, there is much more that can be done in order to expand on this project of clickbait and sentiment analysis. For future work we can try to see if there are any further correlations to clickbait and sentiment analysis. We could perhaps find the polarity score of titles to see if clickbait titles have a more negative and positive polarity score than non-clickbait titles. After all, the purpose of clickbait is to bait online surfers to click on their content. Perhaps sentiment is a clickbait method. Another form of further work would be to expand the amount of data collection to gain a better and more broad view of clickbait and sentiment analysis of clickbait articles. Instead of only using 40 articles, we can go above and beyond and use 100. Unfortunately, one of the biggest challenges of doing a project in this field is advertisement removal. Manually we would need to clean ads so we can focus on only the article at hand. After getting tired of cleaning articles, we bit the dust and copied and pasted all our articles and placed them in a dictionary. Advertisement cleaning was a major challenge when doing this project. Perhaps for future work, we can develop a method that can rip ads clean off from our url's. Without a quick and easy method of removing ad's, there is an unbearable amount of work on our hands. Another future work that can be done is more accuracy testing. We should look to see if our rule based methods are accurate and find a method that can be utilized to test the accuracy of our results and code.

References

[1] Jack Merullo, Luke Yeh, Abram Handler, Alvin Grissom II, Brendan O' Connor, Mohit Iyyer. "Investigating Sports Commentator Bias within a Large Corpus of American Football Broadcasts".

[2] Caoi Butler, Gullal S. Cheema, and Gaurish Thakkar "Combining sentiment analysis classifiers to explore multilingual news articles covering London 2012 and Rio 2016 Olympics" (2022).

[3] Khalid Mahbood, Fayyaz Ali, Hafsa Nimazi."Sentiment Analysis of RSS Feeds on Sports News – A Case Study"(2019).

[4] "https://rdrr.io/cran/sentimentr/man/sentiment.html"

[5] "https://www.red-gate.com/simple-talk/development/data-science-development/sentiment-analysis-python/"

[6] "https://realpython.com/python-nltk-sentiment-analysis/"

[7] "https://www.kaggle.com/datasets/amananandrai/clickbait-dataset"

[8] "https://scikit-learn.org/stable/modules/generated/sklearn.pipeline.Pipeline.html"

[9] "https://stackoverflow.com/questions/2600191/how-do-i-count-the-occurrences-of-a-list-item"

XI. Contribution

Tabshir

In this project, Tabshir was solely responsible for the implementation of the Multinomial Naive Bayes text classifier model that was used to predict the probability of a given article being considered as clickbait. Tabshir created an array that contains at least 200 common words that are hyperbole. In this final paper, he was responsible for the abstract section and everything in the section "V. NAIVE BAYES MODEL".

Rezwan

In this project, Rezwan helped gather 10 of our 40 urls in articleurls.py. Rezwan also put together all the raw data in which he copied and pasted all 40 articles and stored them as a string in rawdata.py. Rezwan also worked on the rule based method of polarity score in PolarityScore.ipynb. Rezwan also worked on everything in DATAandVISUALS.ipynb where he placed all the data in dictionaries, found the polarity score of all the articles, used Tabshir's clickbait score method to get the clickbait score for each title, and built dataframes and bar charts to create visualizations. Wrote previous research, polarity score results, and future work for the final paper.

Nahin

In this project, Nahin helped gather 10 articles from SkySports.com out of the 40 URLS in articleurls.py and put them in an array for everyone to use. Nahin worked on the implementation and algroithim for the rule based model of Hyperbole Score in HyperboleScore.py. Nahin worked in creating the visuals for Hyperbole Score results, implementing the same methods from Rezwans visuals. Nahin also worked on creating the template for the Powerpoint presentation and worked on formatting the paper in Latex.