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
Shezan.Alam48@myhunter

Rezwan Rahman
Hunter College
New York, U.S.A
nahin.mahmood27@myhunter

Abstract—Many studies have shown that video games improve a person's hand-eye coordination, reaction time, and ability to learn new 'sensorimotor' tasks [1]. In addition to this, archery is shown to reduces stress [2]. Unfortunately, not everyone has access to a bow and arrow at home. Luckily, virtual reality (VR) headsets create opportunities to access the same benefits at home [3]. Our project brings the benefits of archery and video games in general into the home of users through virtual reality. The application involves 4 distinct scenes and logic behind bow, arrow, quiver, and target game objects. It uses the XR Interaction Toolkit in the Unity engine as a framework to build a 3D-interactable virtual world.

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

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

VI. Sentiment Analysis - Polarity Score

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)).

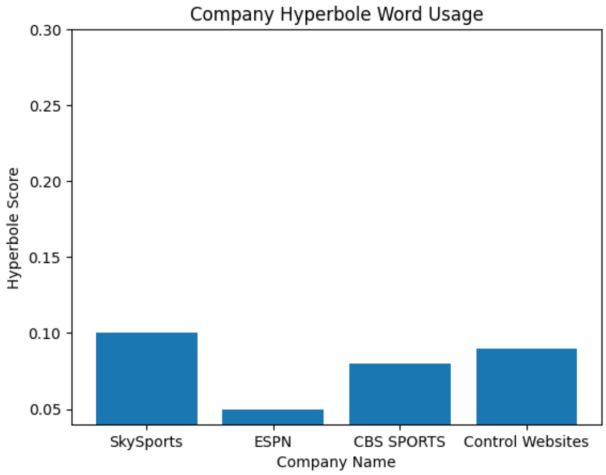
This specifically calls a variable x that calls to check and count for every hyperbole word present in the article text and obtain the sum of all of them. After getting the sum of hyperbole words present, we then divide the sum of the hyperbole words present in the article by the total number of words 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

Sentiment Analysis - Hyperbole

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.

Fig. 3. Results of Hyperbole Word



Based on the results shown above, 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.