Using Article Headlines and Content to Identify Clickbait

Jacob Bettencourt
jacob.bettencourt
@mail.mcgill.ca

Miya Keilin
miya.keilin
@mail.mcgill.ca

Raphaëlle Tseng
raphaelle.tseng
@mail.mcgill.ca

Abstract

In the current age of news cycles and media, people consume more information than ever before. The rapid turnover of stories has led to the exaggeration of headlines and the proliferation of 'clickbait' in order to attract readers and increase page views. Often, these headlines may be misleading, pointing to articles with incongruous content. In this paper, we make use of the Webis Clickbait Corpus 2017 in order to create several pipelines capable of detecting such sensationalist headlines and 'clickbait' articles. We consider a variety of hand chosen features and compare those with features automatically extracted from contemporaries of BERT in order to identify features that may indicate the bias of a headline before applying different models to this data.

1 Introduction

Media organisations have become vastly more reliant on online news platforms to reach audiences in the last dozen years. In 2020, this reliance has been amplified by a global pandemic, forcing print publishers to shut down, and an increasingly turbulent political and social climate that demands faster, more urgent news cycles. Whereas previously, headlines served to inform readers of an article's content, they now also have to attract a reader's attention. As news companies seek to generate more traffic, this has led to the proliferation of sensationalist headlines, misleading clickbait titles, and even outright 'fake news', spreading misinformation in an environment where there is more news available than can possibly be digested by any one person.

The term clickbait is defined as 'something designed to make readers want to click on a hyperlink especially when the link leads to content of dubious value or interest'. We broaden this to incorporate article titles presented in a exaggerated fashion, in an attempt to make information seem more shocking than it really is. Clickbait exists solely online and is dependent on clicks and web page views. It is important to note, however, that we will not be taking into consideration other possible factors that may form part of a clickbait article, such as advertisements or the traffic on a page. Instead, we focus on the text that makes up the headline and the article body, with the goal that this work can be generalized to other related topics such as determining how closely a headline represents its article's content. Some examples of such headlines include CNN's coverage of Ebola: "Ebola in the Air? A Nightmare That Could Happen", and Buzzfeed's '21 Pictures That Will Make You Feel Like You're 99 Years Old'.

Often, clickbait serves to sow uncertainty. Manju Rose Mathews, head of the department of journalism and mass communication at Christ Nagar College, India, stated at an August 2020 UNESCO webinar on "Countering 'Fake News', Misinformation and Sensationalism" that 'false and misleading content is instrumental in shaping public opinion' and 'Developing countries are facing a much greater threat to their democratic process and public opinion building due to the accelerated growth of fake news and misinformation channels. Psychological impacts of these deliberate campaigns on people are catastrophic"[11]. Sensationalist headlines and clickbait pose real dangers for modern journalism, making them worth investigating. We propose novel features that may be used to extract information and better indicate how accurately an article headline represents the article's content. This is a step forward in allowing us to detect clickbait and sensationalist headlines to prevent the spread of misinformation online.

2 Related Work

Clickbait and sensationalist headlines do not necessarily equate 'fake news', a topic which has garnered perhaps even more interest in the last few years. The article content of clickbait is generally

not false; although the titles may be construed as misleading and exaggerated, the articles tend to be genuine.

The first paper to present a machine learning approach to clickbait detection was introduced at the European Conference on Information Retrieval (ECIR) [9]. Potthast, Kopsel, Stein, and Hagen compiled a clickbait corpus of Twitter tweets. They trained a model based on 215 features and first proposed a random forest classifier that achieved 0.76 precision. We build on their idea of extracting relevant features to inform our model. This paper however, divides features into three sections: (1) the teaser message directing people to articles or clickbait, (2) the linked webpage, and (3) meta information attached to the tweet (e.g.image, video, retweets etc.). So whilst we use their research to inform our own, we apply the proposed method on different parts of the corpus with a focus on the text portion of the clickbait i.e. the headline and the article content. A follow up piece, 'Machine Learning Based Detection of Clickbait Posts in Social Media'[1] fed features into 4 different models including a Logistic Regression and a Random Forest Classifier. However, these works do not attempt models such as Naive Bayes, support vector machines (svm), or neural networks. Neural networks are mentioned as being particularly worth exploring.

There has also been work on the specific topic of political leaning detection. 'Detecting Political Bias in News Articles, Using Headline Attention'[4], published in 2019, suggested using an attention mechanism applied on the article based on its headline. This allowed it to attend to more critical content to predict bias.

In 2019, a paper titled 'Learning to Determine the Quality of News Headlines'[7] proposed analysing website logs to determine 4 indicators that could be used to evaluate the quality of headlines. Soft target distribution of the quality indicators, which included click count and dwell time, was used to train a deep learning model to predict the quality of unpublished news headlines. Whilst this paper does consider the semantic relationship between the headline and the body of the article, the focus is placed on website features, and the interaction between readers and web pages. This contrasts with our approach of focusing on the natural language processing (NLP) aspect of the issue, and could be used in conjunction with the results presented below.

It is impossible to look at recent advances in the field of NLP without encountering sophisticated neural network models like BERT which have shown improvements on most NLP tasks, as shown by the improvements on the GLUE and SQuAD scores compared to previous state of the art results[3]. In order to take advantage of the apparent usefulness of these state of the art results we use two contemporaries of BERT. First, we utilize Sentence-BERT to create faster embeddings of text which lend themselves better to comparisons of similarity[10]. Second, we consider the use of the text-summarization model BART in order to simplify the article body so that this can be embedded and used directly in various models which could leverage the summary in its classification[5].

3 Method

In this section, we explain how we developed a method to detect clickbait. We begin by describing the data we used and the features we extracted. We explore working with BART, before presenting the various models we chose to train on the data.

3.1 Data

The data set used is the Webis Clickbait Corpus 2017 [8]. The data is comprised of 38,517 articles collected from Twitter between the first of December, 2016 and the 30th of April, 2017. The authors split the data in half and kept the second testing half separate and so only half of the set was used in these experiments. A maximum of 10 tweets were taken each day in order to prevent over-saturation of one topic. A total of 27 publishers are included in the tweets, which were required to only contain one link and no videos. The publishers chosen were those that were the most retweeted on Twitter. In order to label the data the crowd-sourcing platform Amazon Mechanical Turk was used. Each tweet was annotated by five people on a four point scale, ranging from "Not clickbaiting" to "Heavily clickbaiting." It should be noted that this data set is skewed towards not clickbaiting labels, as can be seen in Figure 1. Each instance has its original HTML, web address, and a web archive file. It should also be noted that the creators of the data set had 459,541 tweets that met the above criteria, and used random sampling to obtain the final set that was labelled.

One of the major difficulties in accumulating

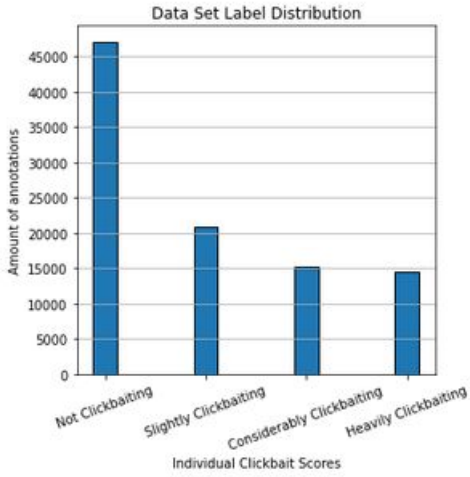

Figure 1: Distribution of the Webis Clickbait Corpus 2017.

web data is that there is no generally accepted standard practice for HTML beyond the required compilation. As such, while it is simple for humans to separate an article's body from its title, advertisements, and the rest of a website, it is not such a simple task for machine parsers. In the experiments listed below the data extracted from the developers of the data set was used, as the main focus of this work is the processing of language. While the provided data occasionally contains additional information such as footer content in the website, on the whole the content is generally extracted quite well.

3.2 Preprocessing and Feature Extraction

The data was preprocessed by removing discourse cues like 'therefore' to extract key information and remove noise. We were reluctant to remove too much information as even stop words could be used as a feature in our method. We chose to implement novel features as well as existing ones already described in recent literature [2]. We began by brainstorming the possible ways a headline and the content of an article might be used to tag an article as clickbait. For example, the aforementioned existing literature proposes sequences and Part-Of-Speech (POS) patterns such as NUMBER + NOUN PHRASE + 'THAT', which would indicate headlines such as '10 things Apple will never tell you about the iPhone'.

We first considered features that were solely reliant on the headline, and independent of the article's body. These included features like the proper use of capitalisation in the title, the use of demonstratives like 'this' and 'that', as well as the inclusion of third-person and second-person pronouns. The features we implemented were features that could be applied to both the headline and the content. Those implemented using the body content include cosine similarity between first sentences in paragraphs, and the headline, the number of contractions, and a series of POS features. Since the data did not come labeled with the POS tags the data was automatically labelled using NLTK's tagger. In future work it would be beneficial to apply a more sophisticated tagger. A comprehensive list of implemented features can be seen in the appendix A.1.

3.2.1 Data Representation

To input the data into the models listed below, it is required to transform the tokenized and preprocessed data into an embedding in vector space. To accomplish this we use the state of the art Sentence-BERT model which is shown to significantly reduce the time required to obtain sentence embeddings that can be used for various similarity metrics[10]. The Sentence-BERT architecture uses a Siamese backbone with two BERT models and additional pooling to obtain vectors which are much more ready to be compared for similarity than the vectors produced by BERT or RoBERTa. Moreover, Sentence-BERT takes a fraction of the time these models require and is shown to maintain the BERT accuracy.

3.2.2 BART

BART is a denoising architecture from Facebook AI which the authors claim is a generalization of the following state of the art models: BERT since it has a bidirectional encoder; GPT (Generative Pre-trained Transformer) due to its left-to-right decoder; various pre-training schemes used in previous models[5]. Not only does BART match RoBERTa on the GLUE and SQuAD benchmarks, but it also shows improvements on the ROUGE[6] performance for text summarizing over other current state of the art methods, making it ideal for our purposes. The BART model here is used in conjunction with the Sentence-BERT word embedding mentioned above. The original article bodies are fed into BART and a summary is generated for the article body. Internally the model only accepts articles of certain lengths, and so the articles were truncated at 4000 characters to accommodate this. This seems to be a reasonable assumption as the

quality of the headline in terms of content quality and sensationalism should be ascertainable within this limit, at least for a human observer. The article embedding is then appended to the title embedding which is fed into a model in order to quantify the clickbait score. It should be noted that if the goal of this data set was to detect how well an article's headline reflects its content, the results of cosine similarity between the embedded headline and article summary could be enough to yield good performance. The issue with using this metric in this context is that a headline can be perceived as clickbait while also having a body that supports the clickbait headline.

3.3 Models

In this section, we talk about our choice to utilise several different models to test the features. We experimented with models already mentioned in existing literature and decided to try new models too. We trained our feature vectors on two neural networks, a naive bayes model or support vector machine, a logistic regression model, and a random forest classifier model.

Both neural networks had three linear layers and used a stochastic gradient descent optimizer. The loss function for the classification model was cross entropy loss and mean square error was used for the regression model. The models were trained with data in batches of size 64.

The data was separated using an 80:20 split. The labels are given in 3 different ways. The first is as the rounded mean score of the 5 annotators: scores greater than or equal to 0.5 were labelled clickbait while scores less 0.5 were not labelled as clickbait. The second set of labels is defined by the nearest label to the mean of the annotators' scores. Finally, we use the mean value of the scores directly for the regression models. The feature vector made up `train_data` and `test_data`.

4 Results

Given that the data set is skewed and most of the articles are not categorized as clickbait, we present the results with a baseline that classifies all the articles as the most frequent label in the training set.

As can be seen from the results in Tables 1, 2, and 3, our methods present a modest improvement over the baseline results in nearly all cases. In terms of accuracy, the neural network model performs best, and the results improve more when the BART summary and Sentence-BERT embeddings are used directly with the neural network in the case of the binary classification problem. The model which performed the best for the 4-class classification problem was the support vector machine with the BART pre-processing. These results should be taken with a grain of salt, as the confusion matrices show in Figures 2 and 3. It is promising that the results of the multi-class predictions label considerably clickbaiting articles, but in both cases the models very rarely classify the data as clickbaiting due to the skewed nature of the data set.

Supervised Model	Accuracy	F1 Score	Precision
2 Classes			
Baseline	76.51	43.35	38.25
Logistic Regression	76.91	76.91	76.91
Naive Bayes	71.29	71.29	71.29
Random Forest Classifier	82.50	82.50	82.50
4 Classes			
Baseline	42.94	15.02	10.73
Logistic Regression	43.76	43.76	43.76
Naive Bayes	38.07	38.07	38.07
Random Forest Classifier	48.92	48.92	48.92

Table 1: Performance of clickbait classifiers using different prediction models (%) compared with a baseline of the most frequent label.

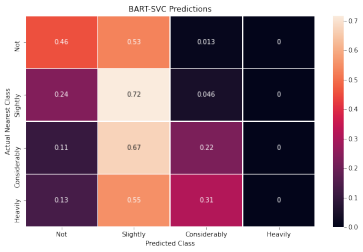

Figure 2: Confusion matrix for the best model with the four class labels.

5 Discussion and Conclusion

In this paper, we proposed a method to identify clickbait by extracting a variety of different features from an article's headline and content. We

Supervised Model	Accuracy	F1 Score	Precision
2 Classes			
Logistic Regression	79.02	43.35	39.10
Random Forest Classifier	77.89	51.53	51.53
Support Vector Machine	80.37	62.73	77.95
4 Classes			
Logistic Regression	46.67	38.35	39.10
Random Forest Classifier	47.26	29.83	37.51
Support Vector Machine	51.20	36.20	39.17

Table 2: Performance of clickbait classifiers using different prediction models with BART preprocessing.

Model	Accuracy(%) for Classification	Loss for Regression
2 Classes		
Neural Net	75.64	0.1874
BART + Neural Net	80.04	18.90
4 Classes		
Classification Neural Net	42.11	1.1932
BART + Neural Net	49.65	1.11

Table 3: Performance of Regression Neural Net

suggest a model that can help classify the quality of news being published from the text portion alone. Our results show an improvement over baseline methods and have resulted in a method which correctly labels an article as clickbait with 80% accuracy and correctly predicts the average label given by human annotators with 51.20% accuracy given the data set. These modest improvements are indicative of the fact that, given a more evenly distributed data set, these methods could show an even greater performance boost over the baseline. In the future, a data set with more labels in each class could reveal the true value of these models, but reducing the size of this particular corpus to adjust the distribution significantly reduces the size of the data available. Given the increasingly important

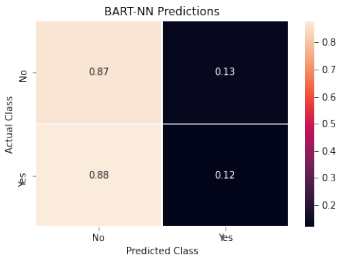

Figure 3: Confusion matrix for the best model with the binary classification labels.

role online news has been playing in determining and informing public opinion, the proposed method could be of key practical value for online news media and the layperson who seeks more transparency from the news they consume.

In future work, performance improvements could be obtained through utilizing more sophisticated Neural Network models, either by directly adding more layers and connections or by implementing a pre-trained model on a related task and adjusting its final layers. Another avenue would be to increase the amount of hand chosen features, with the caveat that too many features could lead to over-fitting of the data and that the process is quite time consuming. Additionally, with the goal of only determining the clickbait score of an article it would be prudent to take into consideration features outside of just the article's text, including the quantity of ads, additional meta data, and some quantifiable form of the layout of the article. As alluded to before, it would be most beneficial to collect more high-quality or high-volume data which is evenly distributed among the various labels in order to approach a generally accepted value for the clickbait score of an article, and so that the results are not skewed towards one end of the distribution.

6 Statement of Contributions

Jacob Bettencourt extracted the data from the set, implemented features, created the BART preprocessing, implemented the text embeddings using Sentence-BERT, and wrote the related sections.

Miya Keilin created the neural network models and implemented a large portion of the features, as well as the required preprocessing for these features and writing the related sections.

Raphaëlle Tseng created the code for the supervised models and wrote much of the report including the abstract, introduction, related work, parts of method, and conclusions.

References

[1] Xinyue Cao, Thai Le, Jason (Jiasheng) Zhang, and Dongwon Lee. Machine learning based detection of clickbait posts in social media. *ArXiv*, 2017.

[2] Xinyue Cao, Thai Le, Jason (Jiasheng) Zhang, and Dongwon Lee. Machine learning based detection of clickbait posts in social media. In *Clickbait Challenge 2017*, 2017.

[3] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. page 4171–4186. Proceedings of NAACL-HLT, June 2019.

[4] Rama Rohit Reddy Gangula, Suma Reddy Duggenpudi, and Radhika Mamidi. Detecting political bias in news articles using headline attention. In *Proceedings of the 2019 ACL Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, 2019.

[5] Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880, Online, July 2020. Association for Computational Linguistics.

[6] Chin-Yew Lin. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain, July 2004. Association for Computational Linguistics.

[7] Amin Omidvar, Hossein Poormodheji, Aijun An, and Gordon Edall. Learning to determine the quality of news headlines. In *12th International Conference on Agents and Artificial Intelligence (ICAART)*, 2020.

[8] Martin Potthast, Tim Gollub, Matthias Hagen, and Benno Stein. The clickbait challenge 2017: Towards a regression model for clickbait strength. *ArXiv*, abs/1812.10847, 2018.

[9] Martin Potthast, Sebastian Kopsel, Benno Stein, and Matthias Hagen. Clickbait detection. In *European Conference on Information Retrieval (ECIR)*, 2016.

[10] Nils Reimers and Iryna Gurevych. Sentence-bert: Sentence embeddings using siamese bert-networks. In *Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing*, pages 3982–3992, November 2019.

[11] Unesco. Countering 'fake news', misinformation and sensationalism, August 2020.

Appendix

A Appendices

A.1 Model Features

Article word count
Number of proper nouns in headline
Number of proper nouns in content
Average length of word in headline
Average length of word in content
Length of longest word in content
Number of question marks in content
Number of contractions in content
Number of contractions in the headline
Headline contains exclamation point
Number of exclamation points in content
Headline contains number
Headline starts with adverb
Average number of words per sentence
Number of superlative adjectives in content
Number of superlative adverbs in content
Number if tokens in content
Stop words to words ratio
Contractions to words ratio
BERT key word extraction cosine similarity with headline
Cosine similarity between headline and each paragraph
Cosine similarity with first sentence in content
First word of headline is who/what/how/why/where

Table 4: Comprehensive list of features used as input to the models. Note that the input features in the BART model were different.

NNP	NNP NNP
IN	NNP VBZ
IN NNP	WRB
NN	Contains QM
PRP	VBZ
NNP NNP VBZ	NN IN
NN IN NNP	NNP .
PRP VBP	WP
DT	NNP IN
IN NNP NNP	POS
IN NN	NNP NNS
IN JJ	NNP POS
WDT	NN NN
NN NN	NN NNP
NNP VBD	RB
NNP NNP NNP	NNP NNP NN
RBS	VBN
VBN IN	Contains Number NP VB
JJ NNP	NNP NN NN
DT NN	Contains EX

Table 5: List of the part of speech (POS) features implemented. All are counts unless otherwise stated. Features with multiple POS tags indicate 2- or 3-gram sequences. These features were applied to both the headline and the article's content separately.