Literature Survey : Identify a clickbait and Un-clickbait it

Team 4: Dama Sravani, Jashn Arora, Tanish Lad and Guru Ravi Shanker
Mentors: Vijayasaradhi Indurthi and Nikhil Pinnaparaju
Professor: Vasudeva Varma

1 Introduction

The influence of social media and the digital world has been increasing day by day. Clickbaits, fake news and others have been increasingly trending in various platforms. These are mainly used by journalists to attract users to view and share their articles on their sites which helps in increasing their revenues. Journalists and other social media sites use clickbaits to exploit the curiosity gap among netizens across the networks and monetize their business model. These are mostly misleading and different from expected by the user, resulting in wastage of time, damages a brand's reputation, and erodes user trust on the sites. So there is a need to identify clickbaits and create a prototype that helps the user choose the correct articles.

Some examples of clickbaits include (as mentioned in Indurthi and Oota (2017)):

- *"21 Completely Engrossing Fan Fictions You Won't Be Able To Stop Reading"*
- *"These White Tiger Cubs Are The Most Beautiful Creatures You'll See Today"*
- *"Here's What Real Vegans Actually Eat"*
- *"Bow Wow Had No Clue How To Kill Time During The Grammys And It Was Hilarious"*
- *"We Know Who Your Celebrity Husband Should Be Based On One Question"*

One of the first studies regarding clickbait dealt with analysing the linguistics aspects of the clickbait (Rowe, 2011; Blom and Hansen, 2015), then features like textual similarity features between the headline and the body, informality and forward reference features, sentence structure features, word pattern features were extracted from the text and were used for classifications (Chakraborty et al., 2016). Nowadays due to the advancements in NLP, RNNs (Anand et al., 2016) and transformer architecture (Indurthi et al., 2020) models are popular. Studies were also done on predicting the intensity of clickbait rather than simple binary classification.

2 Literature Survey

2.1 Predicting Clickbait Strength in Online Social Media

Indurthi et al. (2020) were the first to use transformer-based architecture for predicting the strength or intensity of clickbait on online social media. The authors used the Webis Clickbait Corpus(Clickbait Challenge 2017), consisting of 38517 tweets split equally into training and testing set as the dataset. This was the same dataset mentioned in Zhou (2017) where the tweets are annotated in 4 points Likert scale.

The authors built multiple regressor models using the current state of art word representations and did extensive comparisions. They compared ELMO and Universal encoder word and sentence embeddings, pre-trained models of BERT, GPT, RoBERTa and also fine tuned on the pre-trained models. Different types of regression techniques like Simple Linear Regression (LR), Ridge Regression (RR), Gradient Boosted Regressor (GBR), Random Forest Regression (RFR), Adaboost Regression (ABR) were trained on the above representations.

The challenge ranked the models based on the MSE loss with the mean annotator judgements. Secondary evaluation metrics like Median Absolute Error (MedAE), the F1-Score (F1) and Accuracy (Acc) were also calculated based on the truth class. The results of the fine-tuned models perform better than only pre-trained representations. RoBERTa + GBR model was ranked the best among all other models while testing. With

different regression models, RoBERTa representation achieved the highest F1 and Acc and lowest MedAE and MSE loss. Among the word and sentence embeddings, ELMO + LR achived highest F1 and Acc and Universal Encoder + RR had lowest MedAE and MSE loss in validation set.

2.2 Clickbait Detection in Tweets Using Self-attentive Network

Zhou (2017) aims to evaluate each tweet's level of click-baiting by generating tweets' task-specific vector representations by attending attention to essential tokens.

The dataset provided by Clickbait Challenge 2017 has been used in this paper to train models. While annotating the dataset, for each tweet, four categories were provided to five annotators, which were "not click-baiting", "slightly click-baiting", "consider-ably click-baiting" and "heavily click-baiting" named as "truthJudgments".

The self-attentive network was trained by optimising the cross-entropy loss between the actual annotation distribution and the predicted annotation distribution with hyper-parameter optimisation.The best self-attentive network gave a F1 score of 0.683.

2.3 Clickbait detection using word embeddings

In this paper, Indurthi and Oota (2017) use distributed word representations of the words in the title as features to identify clickbaits in online news media. Their task required them to calculate a clickbait score of each tweet.

They use the clickbait dataset from Potthast et al. (2018). It contained tweets from Twitter. Each tweet in the dataset had the text of the posted tweet and its associated metadata like keywords, time of the post, media linked with the post, description of the target, and the target paragraphs. Despite the tweets' metadata availability, they limit their experiments to only the post's text for training their model.

They use 7 handcrafted features: number of words, number of stopwords, the average word length of clickbait headlines, presence of question form, presence of digits at the beginning of the headline, presence of gerunds, and presence of superlative forms of adjectives. The augment the GloVe embeddings (Pennington et al., 2014) along with these handcrafted features and then take the average of the embeddings of all words in the tweet to get the embedding of the tweet. They model the task as a regression problem and use a linear regression technique to predict the tweet post's clickbait score.

Their methods achieve an F1-score of 64.98% and an MSE of 0.0791, which was higher than the MSE of the baseline system, which was 0.0435. They say that selecting a little complex model or a machine learning technique or with more feature engineering might help improve the model's performance.

2.4 We used Neural Networks to Detect Clickbaits: You won't believe what happened Next!

In this paper, Anand et al. (2016) introduce a Neural Network architecture based on a Recurrent Neural Network for automatically detecting Clickbaits. The model relies on distributed word representations learned from a large unannotated corpus and character embeddings learned via Convolutional Neural Networks.

The input layer of the model transforms each word of the sentence into embedded features, which are a concatenation of word's Distributed word embeddings and character level word embeddings. These word embeddings are then passed to a Bi-directional RNN. The output from the RNN is finally passed through a fully connected neural network with a sigmoid output node that classifies the sentence as clickbait or non-clickbait.

The dataset they used consisted of 15,000 news headlines released by Chakraborty et al. (2016), which has an even distribution of 7,500 clickbait headlines and 7,500 non-clickbait headlines. The non-clickbait headlines in the dataset were sourced from Wikinews, and clickbait headlines were sourced from BuzzFeed, Upworthy, ViralNova, Scoopwhoop, and ViralStories.

The authors experimented with three types of RNN- a simple RNN, LSTM, and GRU and also with three kinds of embedding features consisting of only character-level word embedding (CE), only distributed word's word embedding(WE), and third was the concatenation of both (CE+WE).

It was observed that BiLSTM(CE+WE) model slightly outperforms other models, and the BiLSTM architecture, in general, performs better than BiGRU and BiRNN. Considering the performance of an individual architecture using three different sets of embedding features, the model using a combination of word embeddings and character em-

beddings consistently gave the best results, closely followed by the model with only word embeddings. Their model attained an accuracy of 0.98 with an F1-score of 0.98 and ROC-AUC of 0.99 on the clickbait detection task.

2.5 Clickbait Detection

In this paper, Potthast et al. (2016) propose a new model for clickbait detection. The authors contributed by compiling the first clickbait corpus of 2992 Twitter tweets, 767 of which are clickbait, and developing a clickbait detection model based on 215 features.

For creating the dataset, they selected the top 20 most prolific publishers on Twitter and collected tweets sent by the publishers in week 24 of 2015 that included links. One hundred fifty tweets per publisher were randomly sampled for a total of 2992 tweets. Each tweet was annotated independently by three assessors who rated them as being clickbait or not.

The Clickbait detection model was based on 215 features divided into three categories pertaining to:

- **The teaser message -** The features of the teaser message comprised of basic text statistics like bag-of-words features, sentiment polarity, readability, etc., or dictionary features, where each feature encodes whether or not a tweet contains a word from a given dictionary of specific words or phrases.

- **The linked web page -** These features were based on web pages linked from a tweet, eg. bag-of-words features, a measure of readability, and length of the main content when extracted with Boiler pipe.

- **Meta information -** For eg. tweet sender, whether media is attached with the tweet or not, whether a tweet has been retweeted, the part of day in which the tweet was sent (i.e., morning, afternoon, evening, night).

For evaluation, the corpus was split into datasets for training and testing at a 2:1 training-test ratio. The model was trained using the three well-known learning algorithms logistic regression, naive Bayes, and random forest. To assess detection performance, precision and recall were measured for the clickbait class and the area under the curve (AUC) of the receiver operating characteristic (ROC).

All features combined achieve a ROC-AUC of 0.74 with random forest, 0.72 with logistic regression, and 0.69 with naive Bayes. The precision scores on all features do not differ much across classifiers, the recall ranges from 0.66 with naive Bayes to 0.73 with random forest.

Interestingly, the teaser message features alone compete or even outperform all features combined in terms of precision, recall, and ROC-AUC, using naive Bayes and random forest.

2.6 Stop Clickbait: Detecting and Preventing Clickbaits in Online News Media

Chakraborty et al. (2016) conducted experiments to automatically detect click baits and then build a browser extension which warns the readers about the possibility of being baited by such headlines.

The dataset included 7500 articles from each of click baits and non-click baits. Non-click baits were headlines from Wikinews articles whereas click baits we obtained by crawling websites such as BuzzFeed.

A detailed linguistic analysis is carried out to get insights about semantic and syntactic nuances that occur more frequently in clickbait headlines. While analysing sentence structure, word patterns, language and N-grams were considered, which are used as features in a classification task. Among the different classifiers, SVM performed the best with 0.93 F1 Score.

For Blocking the click baits, approached based on topic similarity and word patterns was implemented. A personalised clickbait classification was also built based on articles the reader has blocked in the past.

2.7 "8 Amazing Secrets for Getting More Clicks": Detecting Clickbaits in News Streams Using Article Informality

In this paper, Biyani et al. (2016) present a machine-learning model to detect clickbaits. They use various features and show that the degree of informality of a webpage (as measured by different metrics) is a strong indicator of it being a clickbait. Given a web page (url, title, and body), their task was to classify it into one of two classes: clickbait or not-clickbait.

Their data comes from different news sites whose pages surfaced on the Yahoo homepage. Sites include the Huffington Post, New York Times, CBS, Associated Press, Forbes, etc. They collected 1349 clickbait and 2724 non-clickbait webpages.

The data came from a period of around one year, covering late 2014 and 2015. The articles covered different domains such as politics, sports, entertainment, science, and finance. The distribution of clickbaits in different categories was: Ambiguous: 68; Exaggeration: 387; Inflammatory: 276; Bait-and-switch: 33; Teasing: 587; Formatting: 185; Wrong: 33; Graphic: 106.

They use several features which include content; similarity between title and the body - less similarity points towards clickbait; informality - clickbait articles tend to be less formal; forward-reference - use of this, that, he, she, etc. as forward references in headlines creates information gap between headline and article spurring curiosity among readers and hence increasing the chance of them clicking; urls - Fetterly et al. (2004) show that urls offer important cues in identifying spam. They use Gradient Boosted Decision Trees to perform their classification experiments, and they also use 5-fold cross-validation.

Their model achieves 75.5% precision and 76.0% recall on the test set. They looked forward to using non-textual features such as images, videos, and user comments on articles as features and also using deep learning to find additional indicators for clickbaits.

References

Ankesh Anand, Tanmoy Chakraborty, and Noseong Park. 2016. We used neural networks to detect clickbaits: You won't believe what happened next! *CoRR*, abs/1612.01340.

P. Biyani, K. Tsioutsiouliklis, and John Blackmer. 2016. "8 amazing secrets for getting more clicks": Detecting clickbaits in news streams using article informality. In *AAAI*.

Jonas Nygaard Blom and Kenneth Reinecke Hansen. 2015. Click bait: Forward-reference as lure in online news headlines. *Journal of Pragmatics*, 76:87–100.

Abhijnan Chakraborty, Bhargavi Paranjape, Sourya Kakarla, and Niloy Ganguly. 2016. Stop clickbait: Detecting and preventing clickbaits in online news media. In *Advances in Social Networks Analysis and Mining (ASONAM), 2016 IEEE/ACM International Conference on*, pages 9–16. IEEE.

Dennis Fetterly, Mark Manasse, and Marc Najork. 2004. Spam, damn spam, and statistics: Using statistical analysis to locate spam web pages. In *Proceedings of the 7th International Workshop on the Web and Databases: Colocated with ACM SIGMOD/PODS 2004*, WebDB '04, page 1–6, New York, NY, USA. Association for Computing Machinery.

Vijayasaradhi Indurthi and Subba Reddy Oota. 2017. Clickbait detection using word embeddings. *CoRR*, abs/1710.02861.

Vijayasaradhi Indurthi, Bakhtiyar Syed, Manish Gupta, and Vasudeva Varma. 2020. Predicting clickbait strength in online social media. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 4835–4846.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, Doha, Qatar. Association for Computational Linguistics.

Martin Potthast, Tim Gollub, Kristof Komlossy, Sebastian Schuster, Matti Wiegmann, Erika Patricia Garces Fernandez, Matthias Hagen, and Benno Stein. 2018. Crowdsourcing a large corpus of clickbait on twitter. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 1498–1507, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

Martin Potthast, Sebastian Köpsel, Benno Stein, and Matthias Hagen. 2016. Clickbait detection. In *Advances in Information Retrieval*, pages 810–817, Cham. Springer International Publishing.

David Rowe. 2011. Obituary for the newspaper? tracking the tabloid. *Journalism*, 12(4):449–466.

Yiwei Zhou. 2017. Clickbait detection in tweets using self-attentive network. *CoRR*, abs/1710.05364.