Detecting Clickbait Titles with Perfect Accuracy using Text Preprocessing and Machine Learning Models Comparison

Erik Zsolt Varga
West University of Timisoara
Timisoara, Romania
Email: erik.varga98@e-uvt.ro

***Abstract*—Clickbait titles, exploiting the so-called Curiosity Gap, intentionally evoke emotions that compel human beings to click, visit, and even linger on various webpages. While some of these clickbait titles are innocuous, others can have a detrimental impact on their readers. The scientific literature has consistently demonstrated the effectiveness of machine learning models in identifying such texts. Therefore, we will be pursuing this approach. In our paper, we evaluate different combinations of preprocessing techniques and train six distinct models, two of which—the BERT and RoBERTa transformers—achieved perfect scoring, achieving an accuracy of 100%.**

***Index Terms*—clickbait title, cross-validation, machine learning.**

I. Introduction

As the internet's popularity grows in a systematic and continuous way, more and more journals and news platforms are moving from physical entities to online ones. These platform sites need monetary gain to function, which in most cases they obtain from the visitors of the webpage. Problems arise when journalists, and as we will see, other content creators as well, intentionally make false claims, partially or fully, use attention-grabbing keywords, or use words that can evoke particularly negative emotions in the readers. These titles are also referred to as clickbait titles. Although until now we have only mentioned journalism, clickbait titles can appear on any online entity, such as tweets or videos. As we will see in Section II, clickbait titles are not limited to journalism.

Our paper examines various machine learning models, testing them multiple times (using cross-validation) and evaluating different metrics. As two of the six constructed models, BERT and RoBERTa, achieved perfect scoring for our dataset, we can conclude that our research can make a significant impact on the detection of clickbait titles.

In Section II we will present three different related works, all aimed at classifying different text into clickbait and non-clickbait categories. This will be followed by Section III, where our proposed approach is presented, providing details about the dataset we used, how the testing was conducted, and a brief theoretical description of the models we used. Section IV provides details about the experiments we conducted and presents the results. Finally, in the last section, Section V, we draw some conclusions and suggest some possible future directions.

II. Related Work

In the [4] paper, the authors emphasize the importance of detecting clickbait titles in online journalism. Hand-crafted features like the presence of question marks or the presence of digits at the beginning of the headline are used in combination with the pre-trained 300-dimensional GloVe embeddings [7] to detect clickbait tweets. This method achieves an MSE of 0.0791 and an F1 score of 64.98%.

In order to predict clickbait entities, not only texts can be used but text can be combined with other features, for example, in the case of YouTube videos, the number of subscriptions of the channel, the number of likes and dislikes, and the number of comments [8]. The author has achieved significant results with different models across several metrics (such as precision, recall, F1 score, or support), all of them being in the 80%-95% range.

Nonetheless, in the [3] paper, three machine learning models have been constructed to classify news headlines into clickbait and non-clickbait categories. The models they have used for classifications are decision trees, random forest, and SVMs. For each model, the accuracy, precision, and recall were all above 90%, while the best model, the SVM, has achieved a scoring of 96%-97%.

III. Proposed Approach

Our proposed method for detecting clickbait titles involves building and fine-tuning several machine learning models to achieve the best possible metrics.

The dataset used to train these models is a balanced one consisting of clickbait and non-clickbait titles. The full dataset contains 32,000 instances, but we have selected only 500 (250 clickbait and 250 non-clickbait) to expedite the training process.

To ensure proper evaluation, we have employed a 5-fold cross-validation procedure for each constructed model. Several averaged metrics were calculated like accuracy, precision, recall and F1 score.

The models were divided into two groups: classical machine learning models and transformers. The classical models include Multinomial Naive Bayes (MNB), Support Vector Machine (SVM), and Logistic Regression (LR). The transformers consist of three models: BERT, DistilBERT, and RoBERTa.

In the following two subsections, we will provide a brief description of each model.

A. Classical Models

- MNB: This probabilistic learning method is based on Bayes' theorem, a well-known statistical concept [5]. Due to its mathematical simplicity, it does not require significant computational power for training, but its applicability is somewhat limited. It is widely used in natural language processing (NLP).
- SVM: An SVM classifies data instances into two groups by defining an optimal hyperplane that maximizes the margin between the two classes [10]. It also employs the One-Class SVM method for outlier detection.
- LR: This statistical model is derived from the famous linear regression model but adapted for classification problems. Its ease of interpretation makes it a popular choice among engineers.

All of the enumerated models require structured data as features, so we need to preprocess the text. For all models, the chosen preprocessing technique is the classical bag-of-words (BoW) approach. BoW involves treating each word in the entire text as an individual feature. If the word appears in the text instances we want to classify, we increase the value of the feature representing the word [1].

B. Transformer Models

The first transformer model was introduced by Google in 2017 [11], and since then, it has demonstrated its effectiveness in various machine learning tasks, especially in the field of NLP. The main strength of transformers lies in their ability to focus on specific parts of their input, a capability called attention.

- BERT: Developed by Google, BERT excels in its ability to read input text sequences from both directions (hence the keyword "bidirectional" in its name) [2]. Its versatility extends from text summarization to question answering and our use case: classification.
- DistilBERT: As its name implies, DistilBERT is a distilled version of BERT. This transformer is a smaller and more lightweight version of the original BERT model, with a 40% reduction in size and 60% faster training speed [9].
- RoBERTa: RoBERTa is a highly optimized version of BERT [6]. It demonstrates the significance of hyperparameters and training data size in enhancing model performance.

Transformers generally require tokenization, but this preprocessing step was fully handled by the high-level Python package "simpletransformers".

IV. Experiments and Results

The TABLE I contains all the Python packages and their versions we have used in the experiments. We provide this information because we believe that reproducibility is a critical component of academic research.

Package	**Version**
pandas	1.5.3
wordcloud	1.9.3
matplotlib	3.7.1
nltk	3.8.1
numpy	1.23.5
sklearn	1.2.2
simpletransformers	0.64.3

TABLE I: Used packages and their versions.

For data analysis purposes, we created two word clouds: one for the most frequently used words in clickbait titles (Fig. 1) and one for the most frequently used words in non-clickbait titles (Fig. 2). Note that these word clouds were not directly used to train the machine learning models.

As discussed in Section III, for the classical models (MNB, SVM, and LR), we employed the BoW preprocessing technique. Before applying the BoW method, we experimented with various combinations of text preprocessing methods, such as lowercase conversion (LC), stopword removal (SR), lemmatization (L), and stemming (S).

The performance metrics for these combinations are shown in TABLES II, III, and IV. As evident, for all classical models, the best results were obtained without using any text preprocessing methods beyond the mandatory BoW transformation, which is essential for feeding text data into the machine learning models. While all models achieved high scores, the MNB (without any preprocessing) achieved the highest accuracy of 95%.

Accuracy	Precision	Recall	F1 score	Preprocessings
0.950	**0.919**	**0.988**	**0.952**	**Nothing**
0.940	0.921	0.964	0.941	LC + SR
0.936	0.923	0.952	0.937	L + LC + SR
0.922	0.903	0.948	0.924	S + LC + SR
0.922	0.902	0.948	0.924	L + S + LC + SR

TABLE II: Accuracy, precision, recall and F1 score for the MNB model with different preprocessings. Please refer to the third paragraph of Section IV for the abbreviations of the preprocessing methods.

Accuracy	Precision	Recall	F1 score	Preprocessings
0.930	**0.955**	**0.904**	**0.928**	**Nothing**
0.872	0.848	0.908	0.877	LC + SR
0.890	0.866	0.924	0.894	L + LC + SR
0.886	0.876	0.900	0.888	S + LC + SR
0.898	0.891	0.908	0.899	L + S + LC + SR

TABLE III: Accuracy, precision, recall and F1 score for the SVC model with different preprocessings. Please refer to the third paragraph of Section IV for the abbreviations of the preprocessing methods.

Regarding the transformer models (BERT, DistilBERT, and RoBERTa), no predefined preprocessing was required, although tokenization was necessary as it is with most trans-

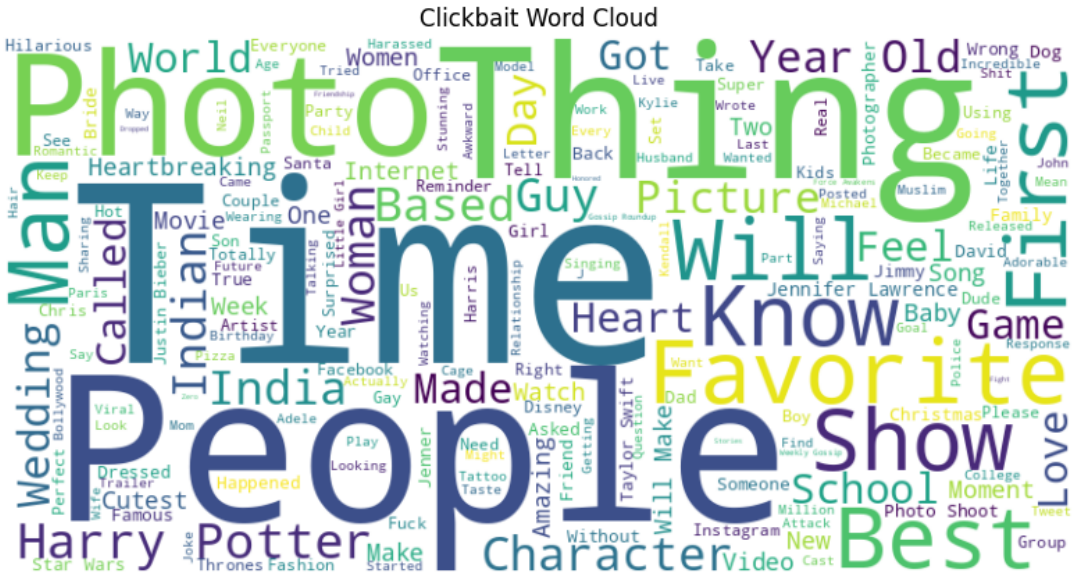

Fig. 1: The most common words used in clickbait titles.

Fig. 2: The most common words used in non-clickbait titles.

Accuracy	Precision	Recall	F1 score	Preprocessings
0.938	**0.944**	**0.932**	**0.937**	**Nothing**
0.932	0.901	0.972	0.935	LC + SR
0.928	0.913	0.948	0.930	L + LC + SR
0.920	0.910	0.932	0.921	S + LC + SR
0.930	0.929	0.932	0.930	L + S + LC + SR

TABLE IV: Accuracy, precision, recall and F1 score for the LR model with different preprocessings. Please refer to the third paragraph of Section IV for the abbreviations of the preprocessing methods.

former models. However, the tokenization step was fully handled by the Python package we used. The results for the transformer models are presented in Table V. As can be observed, BERT and RoBERTa produced perfect results, reaching an exceptional 100% accuracy, while DistilBERT closely followed with 99.8% (mistaking only two predictions across two different folds during cross-validation).

The complete visual comparison of the accuracies of all the models can be found in Figure 3.

It is important to note that all these results were obtained using 5-fold cross-validation to ensure an accurate assessment of the models' performance.

Accuracy	Precision	Recall	F1 score	Model
1.000	**1.000**	**1.000**	**1.000**	**BERT**
0.998	1.000	0.996	0.998	DistilBERT
1.000	**1.000**	**1.000**	**1.000**	**RoBERTa**

TABLE V: Accuracy, precision, recall and F1 score for the transformer models.

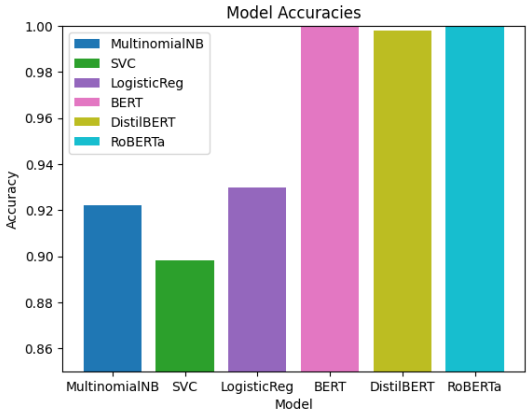

Fig. 3: The accuracy of all the models, ranging from 85% to 100% on the Y-axis.

V. Conclusion

The primary conclusion drawn from our experiments serves as a reinforcement of the assertions made in Section III: the emergence of transformers has cemented their position as the most effective models, particularly for text-specific tasks. Our attainment of 100% accuracy using cross-validation for two distinct transformers corroborates this statement. It is noteworthy to mention that the classical model also yielded favorable results, with all accuracies exceeding 85%.

As a future direction, evaluating the aforementioned models using the same metrics on the comprehensive dataset encompassing 32,000 instances could enhance their robustness and generalization, even if it results in a slight decline in accuracy. Additionally, exploring hyperparameter optimization for classical models would be an intriguing direction, given that training such simpler models compared to transformers requires less time.

References

[1] Gobinda G Chowdhury. *Introduction to modern information retrieval*. Facet publishing, 2010.

[2] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

[3] Ruchira Gothankar, Fabio Di Troia, and Mark Stamp. Clickbait detection for youtube videos. In *Artificial Intelligence for Cybersecurity*, pages 261–284. Springer, 2022.

[4] Vijayasaradhi Indurthi and Subba Reddy Oota. Clickbait detection using word embeddings. *arXiv preprint arXiv:1710.02861*, 2017.

[5] Ashraf M Kibriya, Eibe Frank, Bernhard Pfahringer, and Geoffrey Holmes. Multinomial naive bayes for text categorization revisited. In *AI 2004: Advances in Artificial Intelligence: 17th Australian Joint Conference on Artificial Intelligence, Cairns, Australia, December 4-6, 2004. Proceedings 17*, pages 488–499. Springer, 2005.

[6] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*, 2019.

[7] Jeffrey Pennington, Richard Socher, and Christopher D Manning. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543, 2014.

[8] Abinash Pujahari and Dilip Singh Sisodia. Clickbait detection using multiple categorisation techniques. *Journal of Information Science*, 47(1):118–128, 2021.

[9] Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. *arXiv preprint arXiv:1910.01108*, 2019.

[10] Bernhard Schölkopf, Alex J Smola, Robert C Williamson, and Peter L Bartlett. New support vector algorithms. *Neural computation*, 12(5):1207–1245, 2000.

[11] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.