Francis Wilde at SemEval-2023 Task 5: Clickbait Spoiler Type Identification with Transformers

Vijayasaradhi Indurthi and **Vasudeva Varma**
Information Retrieval and Extraction Lab
International Institute of Information Technology, Hyderabad

vijayasaradhi.i@research.iiit.ac.in and vv@iiit.ac.in

Abstract

Clickbait is the text or a thumbnail image that entices the user to click the accompanying link. Clickbait posts hide the critical elements of the article and reveal partial information in the title, which arouses sufficient curiosity and motivates the user to click the link. In this work, we identify the kind of spoiler given a clickbait title. We formulate this as a text classification problem. We finetune pretrained transformer models on the title of the post and build models for the clickbait-spoiler classification. We achieve a balanced accuracy of 0.70 which is close to the baseline.

1 Introduction

Clickbait-spoiling (Hagen et al., 2022) aims to counteract the manipulative tactics used by clickbait creators and to help people avoid wasting their time on low-quality content. This is achieved by revealing the "suspense" that has been deliberately created in the title by the authors of the post. Without clickbait spoiling, the reader has to spend considerable time to satisfy his curiosity by finding the content which was promised in the title. By clickbait-spoiling, the user can easily skim through the key content without wasting valuable time.

The Clickbait Spoiling task at SemEval 2023 (Fröbe et al., 2023a) consists of two subtasks, Spoiler Type Classification and Spoiler Generation.

- **Spoiler Type Classification:** In this task, the goal is to classify a given clickbait post into one of the three kinds of spoiler types i.e phrase, passage or multi depending on the kind of text required to spoil the clickbait.
- **Spoiler Generation:** In this task, the goal is to generate the actual spoiler for the clickbait post.

Both tasks are presented in the English language. We attempted the first task which is Spoiler Type Classification. We use pre-trained transformer models and finetune them on the postText to train a spoiler-type prediction model. We also experiment with pretrained transformer representations and train classical ML models on these representations for classifying the spoiler type. Our final model, a vanilla transformer model finetuned on this dataset, achieves a balanced accuracy metric of 0.7 which is close to the baseline.

2 Background

In the task for Clickbait Spoiler Type classification, the input is a tweet along with its associated metadata. Table 1 lists the various fields available from the dataset.

Table 2 lists few examples of clickbaits and their spoiler types.

We formulate the problem of spoiler type identification as a text classification task. We hypothesize that a human can easily predict the spoiler type of a clickbait by looking at the post of the clickbait i.e in our case, it is the postText associated with the tweet. We adopt the method followed by (Indurthi et al., 2020) for detecting the clickbait intensity where the authors use only the postText field to predict the intensity of the clickbait. We also limit ourselves to using only the postText field in the dataset.

We train models to predict the spoiler type given the postText of the tweet. We trained models with the combination of postText and targetParagraphs but they performed worse compared to the postText alone. Hence we settled to using only the postText for the task.

3 Transformers

Transfer learning is a popular approach in Natural Language Processing that involves leveraging knowledge gained from one task to improve performance on another related task. The transformer architecture introduced by (Vaswani et al., 2017)

Proceedings of the The 17th International Workshop on Semantic Evaluation (SemEval-2023), pages 1890–1893
July 13-14, 2023 ©2023 Association for Computational Linguistics

Table 1: Fields available in the Clickbait Spoiling Corpus

Field Name	**Field Description**
uuid	The uuid of the dataset entry.
postText	The text of the clickbait post which is to be spoiled
targetTitle	The title of the linked web page to classify the spoiler type
targetUrl	The URL of the linked web page.
humanSpoiler	The spoiler generated by the human
spoiler	The extracted spoiler for the clickbait post
tags	The spoiler type ("phrase", "passage" or "multi") that has to be classified

Table 2: Some examples of clickbaits and their spoiler types

Title	**Spoiler-Type**
The perfect way to cook rice so that it's perfectly fluffy and NEVER sticks to the pan	phrase
You're probably missing out on this major way to save money	phrase
China invited a reporter to hit their new glass bridge with a sledgehammer to prove it's safe. He proved something.	passage
Instagram Just Killed This Feature	passage
Doctors reveal the 5 most common Pokémon Go injuries – and how to avoid them	multi
11 Simple Weight Loss Strategies For Fruitful Results	multi

has become the backbone of many state-of-the-art NLP models. Transformers employ a self-attention mechanism to process input sequences and capture the relationships between different parts of the sequence. This architecture has proven highly effective in capturing complex patterns in text data, enabling models to achieve impressive performance on various NLP tasks (Wang et al., 2019b) and (Wang et al., 2019a). Pretrained models such as BERT (Devlin et al., 2018), DistilBERT (Sanh et al., 2019), RoBERTa (Liu et al., 2019), and DistilRoBERTa were trained on massive amounts of text data. They can be fine-tuned on specific tasks with relatively little additional data, resulting in significant performance gains. DistilBERT, a smaller and faster variant of BERT, has shown comparable performance on many tasks. In contrast, RoBERTa and DistilRoBERTa, trained with improved techniques, have achieved even higher performance on several benchmarks. These pre-trained transformer models have revolutionized the field of NLP and enabled researchers to achieve state-of-the-art results on various tasks.

Table 3: Train/Dev/Test splits

Split	Count	phrase	passage	multi
Train	3200	1367	1274	559
Develop	800	335	322	143
Test	1000	NA	NA	NA

4 System Overview

We follow the standard way of funetuning transformer models where pretrained transformer models are finetuned. In this method, all the layers of the transformer model and the weights in the classification layer are jointly learned during the finetuning process. We experiment with 4 transformer models i.e BERT, RoBERTa, DistilBERT and DistilRoBERTa. We use only the postText field to do the fine-tuning. We do not do any specific preprocessing. We finetune each model for 3 epochs. We use a learning rate of 2e-5. Default values provided by the huggingface library are used for the rest of the configurable parameters.

5 Experimental Setup

The organizers of the task have compiled a dataset of 5000 clickbaits. Out of this 1000 samples are reserved as test data and their ground truths were kept hidden from the participants. The remaining 4000 samples were released for training the models. Out of these 4000 samples, we used 3200 samples for training our models and kept the other 800 samples for development/validation.

We use the huggingface transformers library [1] for training our models.

Our metric for evaluation is the accuracy and the F1-score. Our training and evaluation was performed on Google Colab [2].

The clickbait evaluation happens on a platform called TIRA (Fröbe et al., 2023b) [3]. The participants can either submit a docker image or upload code and models on a virtual machine hosted on tira, which can read the test data and produce the results. The TIRA platform is used so that the test data is kept confidential and to prevent participants overfitting to the test data. In addition, the participants can also upload the results file on the web and get the results evaluated.

6 Results

Table 4 show the performance of the models on the development/validation set of 800 tweets. We can observe that RoBERTa model has performed much better in terms of the Accuracy, Precision, Recall and F1 compared to all other models. RoBERTa achieves an accuracy of 0.72 and an F1 of 0.71 which is very close to the baseline performance. Hence we submitted the results produced by this model on the official test set for the final official evaluation. Table 5 shows the result of the model on the official test set. We can see that on the test set, RoBERTa model produces an accuracy of 0.70 and comparable F1 scores for all the classes.

7 Conclusion

The approach followed by us is a very simple method by just using the postTitle. In the future we can experiment with more fields and leverage semi-supervised learning techniques and improve the classification performance. We can explore more transformer models also.

[1] https://github.com/huggingface/transformers
[2] https://colab.google.com
[3] https://tira.io

Model	Huggingface Model Identifier	Accuracy	Precision	Recall	F1
BERT	bert-base-uncased	0.6388	0.6855	0.5898	0.6099
DistilBERT	distilbert-base-uncased	0.6200	0.7062	0.5761	0.5964
RoBERTa	roberta-base	**0.7200**	**0.7648**	**0.6834**	**0.7068**
DistilRoBERTa	distilroberta-base	0.6913	0.7615	0.6548	0.6780

Table 4: Results on the development/validation dataset of 800 clickbaits

Table 5: Overview of the effectiveness in spoiler type prediction (subtask 1 at SemEval 2023 Task 5) measured as balanced accuracy over all three spoiler types and precision (Pr.), recall (Rec.), and F1 score (F1) for phrase, passage, and multi spoilers on the test set. We report all runs by Team pan23-francis-wilde.

Submission			Accuracy	Phrase			Passage			Multi		
Team	Approach	Run		Pr.	Rec.	F1	Pr.	Rec.	F1	Pr.	Rec.	F1
pan23-francis-wilde	upload	2023-01-25-04-55-22	0.70	0.74	0.69	0.71	0.68	0.78	0.73	0.74	0.64	0.69

References

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

Maik Fröbe, Tim Gollub, Matthias Hagen, and Martin Potthast. 2023a. SemEval-2023 Task 5: Clickbait Spoiling. In *17th International Workshop on Semantic Evaluation (SemEval-2023)*.

Maik Fröbe, Matti Wiegmann, Nikolay Kolyada, Bastian Grahm, Theresa Elstner, Frank Loebe, Matthias Hagen, Benno Stein, and Martin Potthast. 2023b. Continuous Integration for Reproducible Shared Tasks with TIRA.io. In *Advances in Information Retrieval. 45th European Conference on IR Research (ECIR 2023)*, Lecture Notes in Computer Science, Berlin Heidelberg New York. Springer.

Matthias Hagen, Maik Fröbe, Artur Jurk, and Martin Potthast. 2022. Clickbait Spoiling via Question Answering and Passage Retrieval. In *60th Annual Meeting of the Association for Computational Linguistics (ACL 2022)*, pages 7025–7036. Association for Computational Linguistics.

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, Barcelona, Spain (Online). International Committee on Computational Linguistics.

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

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

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *NIPS*, pages 5998–6008.

Alex Wang, Yada Pruksachatkun, Nikita Nangia, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R. Bowman. 2019a. SuperGLUE: A stickier benchmark for general-purpose language understanding systems. *arXiv preprint 1905.00537*.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R. Bowman. 2019b. GLUE: A multi-task benchmark and analysis platform for natural language understanding. In *ICLR*.